Crop Recommendation for India

Tasotti Kerstin

2022-03-08

1. INDRODUCTION

I decided to analyze a crop recommendation dataset as my own submission project for the edX course HarvardX PH125.9x - Data Science:Capstone Course. The aim of this project is to demonstrate the acquired skills in R programming and their analysis in a real world datasets. This dataset is to recommend optimum crops to be cultivated by farmers based on several parameters and help them make an informed decision before cultivation. The data used in this project is loaded from www.kaggle.com. and contains data from the Indian agriculture. This data is relatively simple with only 7 soil and environmental conditions. The data contains Nitrogen, Phosphorus, Potassium and pH values of the soil. The environmental conditions are the humidity, temperature and rainfall which are collected for a particular crop.In the dataset is no information on the harvest yield. Precision agriculture is in trend nowadays. It helps the farmers to get informed decision about the farming strategy. Here, I present you a dataset which would allow the users to build a predictive model to recommend the most suitable crops to grow in a particular farm based on various parameters.

Data fields: Soil condition N - ratio of Nitrogen content in soil in ppm P - ratio of Phosphorus content in soil in ppm K - ratio of Potassium content in soil in ppm
ph - pH value of the soil

Environment condition temperature - temperature in degrees Celsius humidity - relative humidity in % rainfall - rainfall in mm

A recommendation system is a subclass of information filtering system that seeks to predict the “rating” or “preference” a user would give to an item. Recommendation systems is one of the most used machine learning algorithms and will be used in nearly all different areas of our life (Trading, Hospitality, Travelling,…). Companies like Amazon use these systems to learn more about their customer and provide them with products more effectively.

This dataset is prepared, and different analysis were done to develop an algorithm of a machine learning which can predict crop rate and calculate the RMSE (Root mean square error). The RMSE is a KPI to measure the differences between the predicted values of a model and the actual values seen in the data. Therefore the dataset is split into a training set (train) and a final hold-out test set (validation). The objective was for the final algorithm to predict ratings with a root mean square error (RMSE)less as possible. We want to generate a nearly accurate machine learning algorithm.

2. DATASET

For this project we focus on the Kaggle dataset collected by Atharva Ingle and it can be found on Kaggle website (https://www.kaggle.com/atharvaingle/crop-recommendation-dataset (https://www.kaggle.com/atharvaingle/crop-recommendation-dataset)). The data is uploaded in GitHub and loaded into the model.

2.1. DATALOAD

```
`##########################################################################################`

# Load of full, train set, validation set (final hold-out test set)

`##########################################################################################`


if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")
if(!require(ggplot2)) install.packages("ggplot2", repos = "http://cran.us.r-project.org")
if(!require(Metrics)) install.packages("Metrics", repos = "http://cran.us.r-project.org")
if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")
if(!require(data.table)) install.packages("data.table", repos = "http://cran.us.r-project.org")
if(!require(readr)) install.packages("readr", repos = "http://cran.us.r-project.org")
if(!require(tinytex)) install.packages("tinytex", repos = "http://cran.us.r-project.org")
if(!require(knitr)) install.packages("knitr", repos = "http://cran.us.r-project.org")

library(tidyverse)
library(ggplot2)
library(Metrics)
library(caret)
library(data.table)
library(readr)
library(tinytex)
library(knitr)

## Crop dataset:

### <https://www.kaggle.com/atharvaingle/crop-recommendation-dataset>

### <https://www.kaggle.com/atharvaingle/crop-recommendation-dataset/download>

# Crop dataset:
# https://github.com/KerstinTasotti/Crop.git


###Full dataset:
Crop_rec <- read.csv("https://raw.githubusercontent.com/KerstinTasotti/Crop/main/Crop_recommenda
tion.csv", header=TRUE,stringsAsFactors = FALSE)

### Validation set will be 10% of Crop data
set.seed(1, sample.kind="Rounding")
train_index <- createDataPartition(y=Crop_rec$label, p=0.9, list=FALSE, time=1)
train<- Crop_rec[train_index,]
validation <- Crop_rec[-train_index,]
```

3. ANALYSIS OF THE DATA

All analysis in this section will be done with the training set (train). The validation set will be used for the final test of the developed algorithm. The dataset is shuffled and the first 1980 lines of the 2200 lines will be taken for the training set.

At first the structure of the dataset will be analyzed to get familiar with it. The data set contains 2200 rows and 8 variables (N, P, K, temperature, humidity, ph, rainfall, label). The train set contains 1980 rows and the same 8 variables.

```
str(train)

head(train) %>% print.data.frame
```

Each label has 90 entries. Therefore, we select for further calculations m=90.

```
train %>% group_by(label) %>% summarise(count=n()) %>% arrange(desc(count))
m <- 90
```

In the summary we can see the min, max and mean values of the different ground condition.

```
summary(train)

       N                P                K           temperature
 Min.   :  0.00   Min.   :  5.00   Min.   :  5.00   Min.   : 8.826
 1st Qu.: 21.00   1st Qu.: 28.00   1st Qu.: 20.00   1st Qu.:22.782
 Median : 37.00   Median : 51.00   Median : 32.00   Median :25.629
 Mean   : 50.52   Mean   : 53.36   Mean   : 48.14   Mean   :25.670
 3rd Qu.: 84.00   3rd Qu.: 68.00   3rd Qu.: 49.00   3rd Qu.:28.614
 Max.   :140.00   Max.   :145.00   Max.   :205.00   Max.   :43.675


 humidity              ph            rainfall          label
Min.   :14.26   Min.   :3.505   Min.   : 20.36   Length:1980
1st Qu.:60.37   1st Qu.:5.969   1st Qu.: 64.51   Class :character
Median :80.47   Median :6.420   Median : 95.03   Mode  :character
Mean   :71.50   Mean   :6.468   Mean   :103.49
3rd Qu.:89.92   3rd Qu.:6.921   3rd Qu.:124.22
Max.   :99.98   Max.   :9.935   Max.   :298.56
```

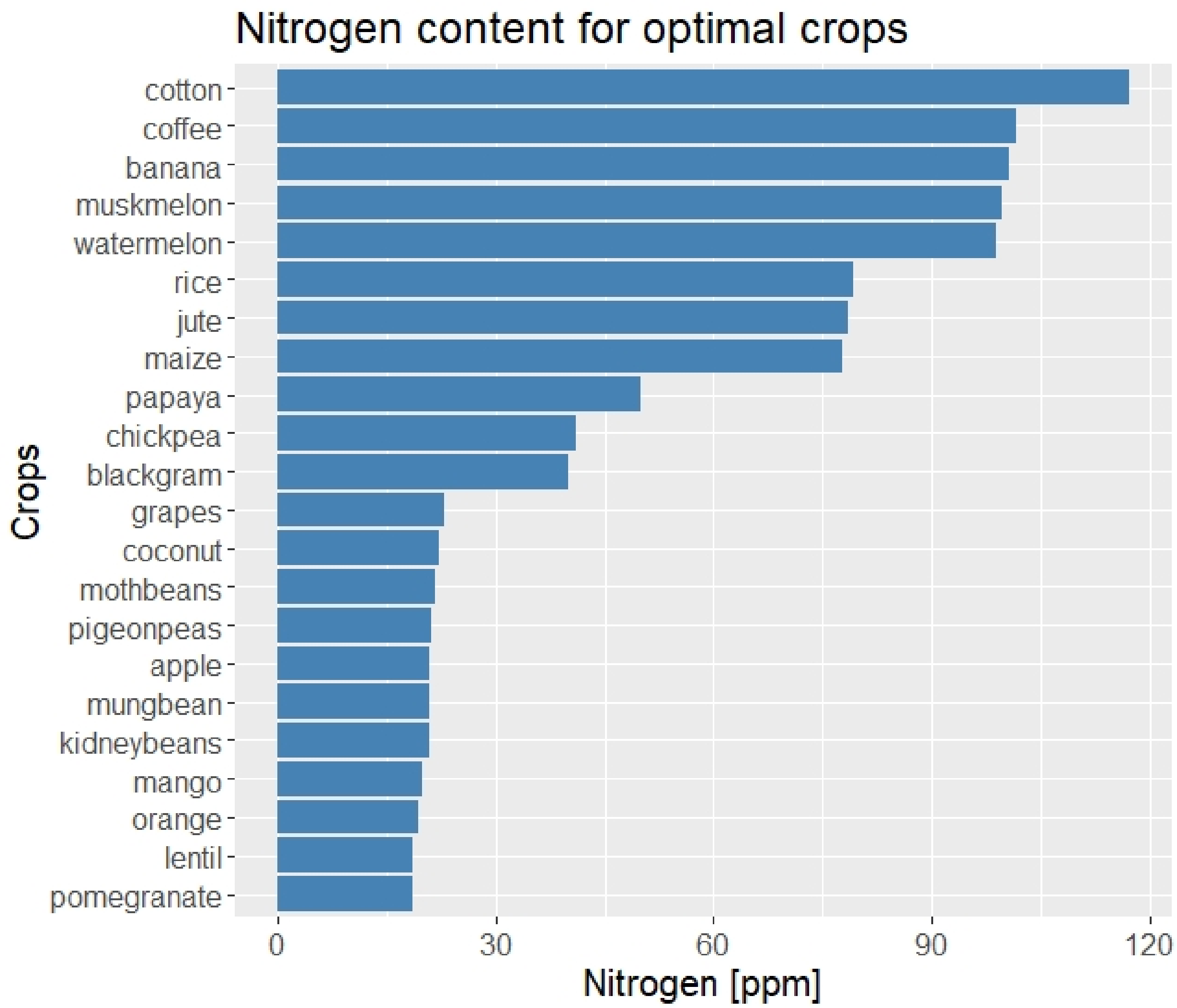
There are 22 different labels in the dataset:

```
unique(train$label)

[1] "rice"        "maize"       "chickpea"    "kidneybeans" "pigeonpeas"  "mothbeans"   "mungbea
n"    "blackgram"   "lentil"      "pomegranate"
[11] "banana"      "mango"       "grapes"      "watermelon"  "muskmelon"   "apple"       "orang
e"      "papaya"      "coconut"     "cotton"
[21] "jute"        "coffee"
```

Nitrogen

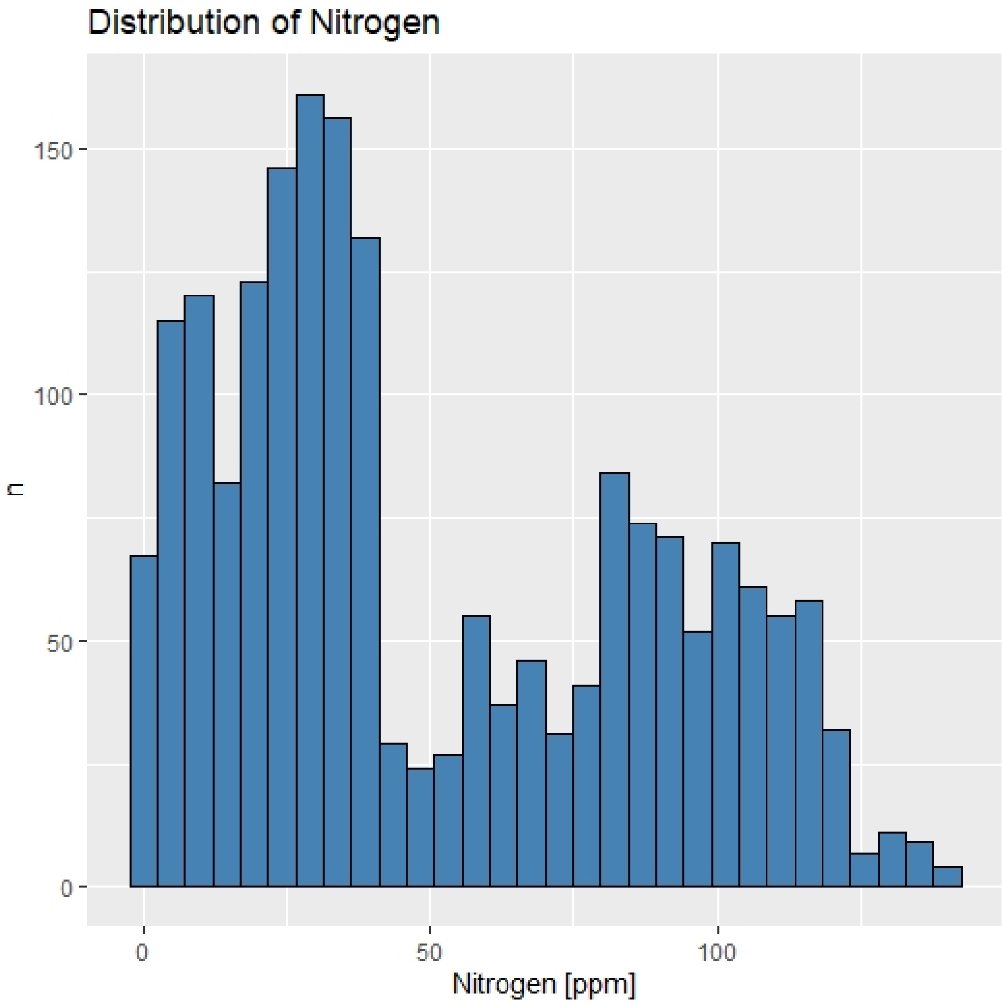
Nitrogen is considered the most important component for supporting plant growth. Nitrogen is part of the chlorophyll molecule, which gives plants their green color and is involved in creating food for the plant through photosynthesis. Lack of nitrogen shows up as general yellowing (chlorosis) of the plant. The mean nitrogen value is 50.5ppm, whereby cotton and coffee need the most nitrogen and lentil and pomegranate need the lowest nitrogen value. A list of all mean Nitrogen values is summarized in the following table and bar chart:

```
train %>%group_by(label)%>%summarise(mean(N)) %>% print.data.frame()

         label   mean(N)
1        apple  21.00000
2       banana 100.78889
3    blackgram  39.95556
4     chickpea  40.92222
5      coconut  22.06667
6       coffee 101.64444
7       cotton 117.28889
8       grapes  22.96667
9         jute  78.40000
10 kidneybeans  20.88889
11      lentil  18.67778
12       maize  77.75556
13       mango  19.84444
14   mothbeans  21.72222
15    mungbean  20.88889
16   muskmelon  99.74444
17      orange  19.42222
18      papaya  49.82222
19  pigeonpeas  21.11111
20 pomegranate  18.64444
21        rice  79.17778
22  watermelon  98.78889
```

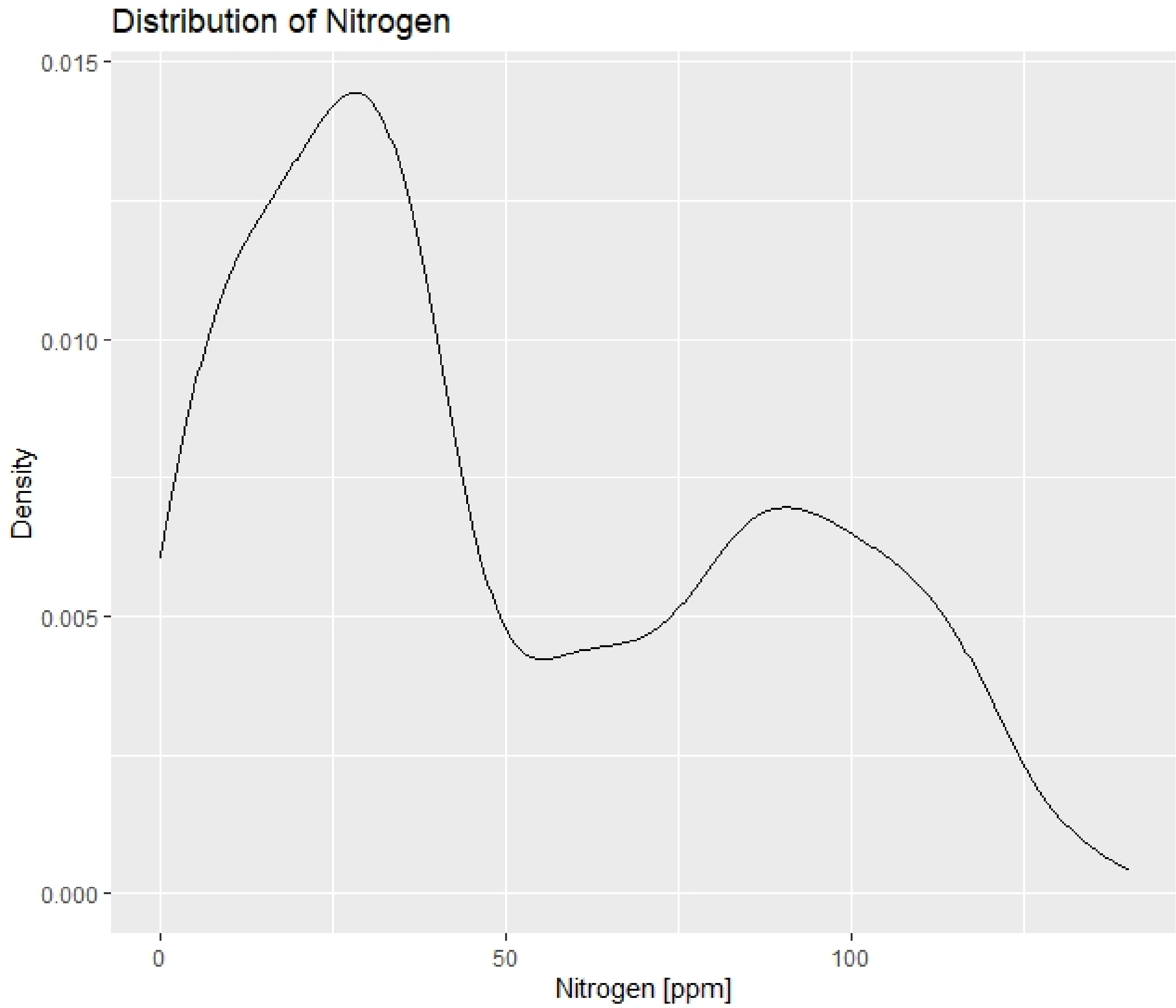


```
train %>% group_by(label) %>% ggplot(aes(x=reorder(label,N/m),y=N/m)) + geom_bar(stat = "identit
y",  fill="steelblue") +labs(x="Crops", y="Nitrogen [ppm]", title="Nitrogen content for optimal
crops") +  theme(text = element_text(size=15))+coord_flip()
```

The Histogram and Density chart shows the distribution of Nitrogen. The most crops prefer a soil with lower Nitrogen content.




```
ggplot(train,aes(N)) +geom_histogram(bins=30, fill="steelblue",color="black")+labs(x="Nitrogen
[ppm]",y="n")+ggtitle("Distribution of Nitrogen")
```

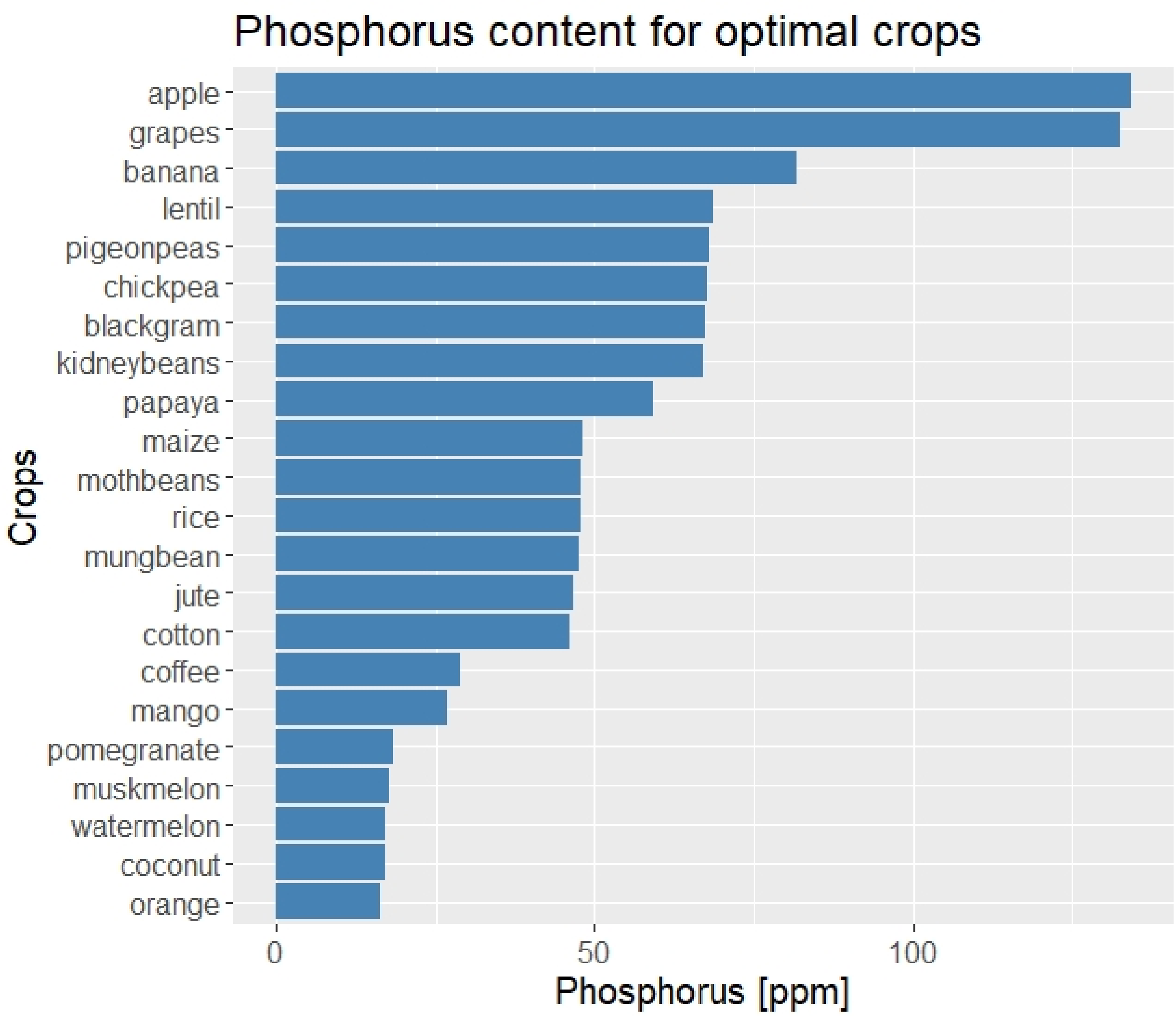



```
ggplot(train,aes(N)) +geom_density()+labs(x="Nitrogen [ppm]",y="Density")+ggtitle("Distribution
of Nitrogen")
```

Phosphorus

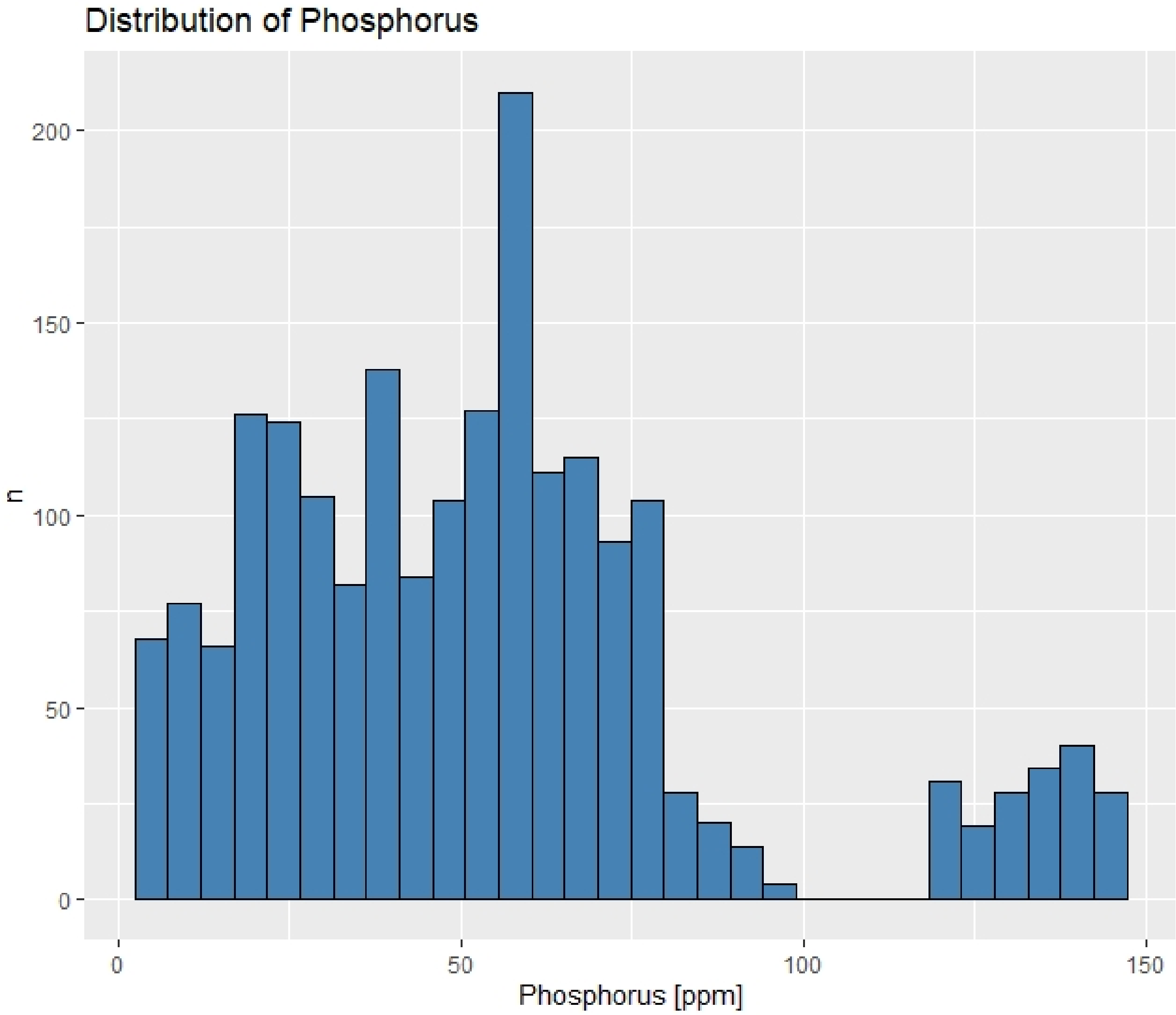
Phosphorus is, therefore, important in cell division and development of new tissue. Phosphorus is also associated with complex energy transformations in the plant. Adding phosphorus to soil low in available phosphorus promotes root growth and winter hardiness, stimulates tillering, and often hastens maturity. Apple and grapes need ground with the highest phosphorus content and orange and coconut needs the lowest content. A list of all mean Phosphorus values is summarized in the following table and bar chart:

```
train %>%group_by(label)%>%summarise(mean(P)) %>% print.data.frame()

        label   mean(P)
1       apple 134.34444
2      banana  81.87778
3   blackgram  67.46667
4    chickpea  67.66667
5     coconut  17.11111
6      coffee  28.78889
7      cotton  46.18889
8      grapes 132.45556
9        jute  46.82222
10 kidneybeans  67.15556
11     lentil  68.54444
12      maize  48.17778
13      mango  26.86667
14  mothbeans  47.92222
15   mungbean  47.64444
16  muskmelon  17.72222
17     orange  16.45556
18     papaya  59.31111
19 pigeonpeas  67.93333
20 pomegranate 18.47778
21       rice  47.85556
22 watermelon  17.22222
```

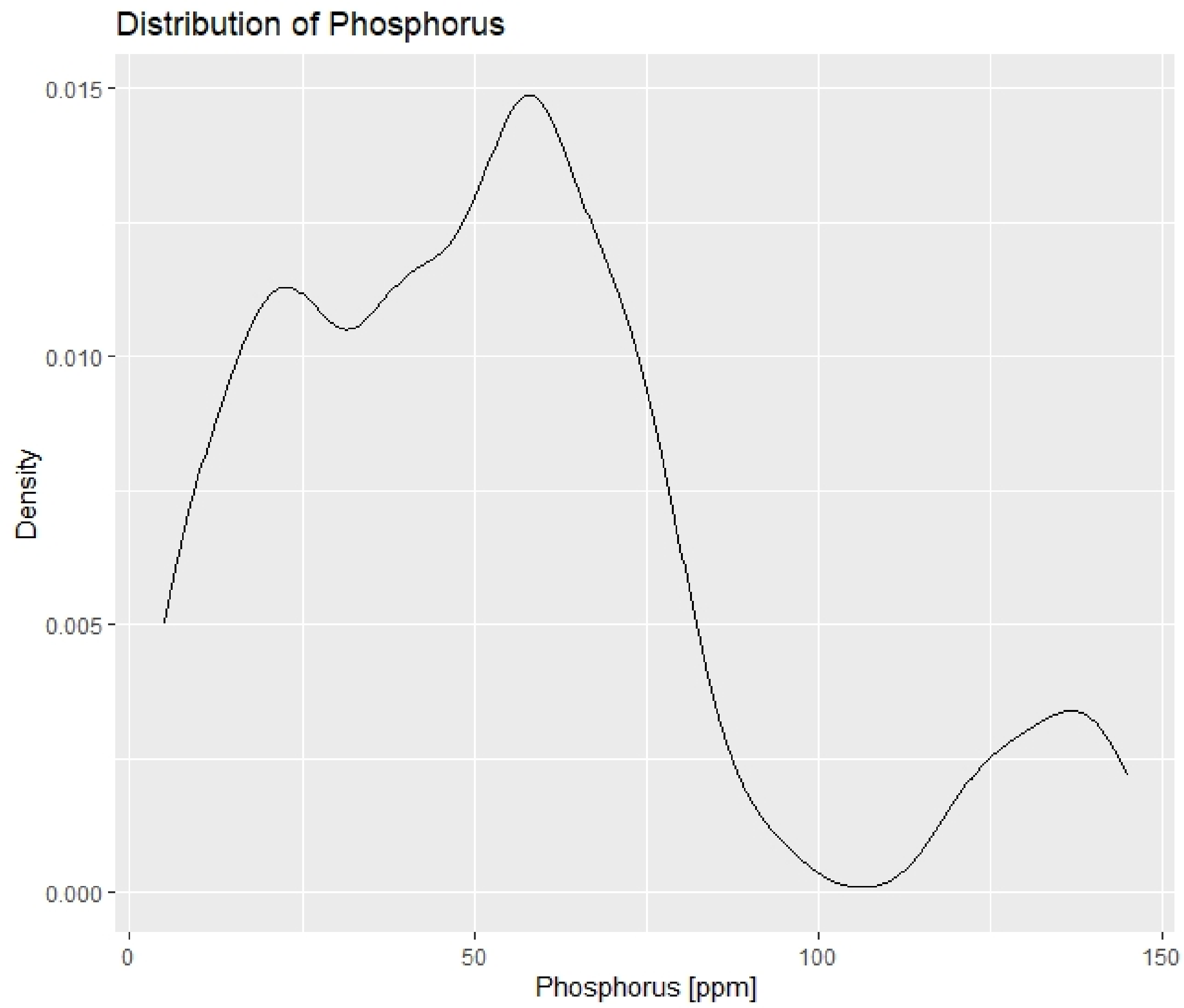


```
train %>% group_by(label) %>% ggplot(aes(x=reorder(label,P/m),y=P/m)) + geom_bar(stat = "identit
y",  fill="steelblue") +labs(x="Crops", y="Phosphorus [ppm]", title="Phosphorus content for opti
mal crops") +  theme(text = element_text(size=15))+coord_flip()
```

The Histogram and Density chart shows the distribution of Phosphorus. Only grapes and apples prefer a soil with a Phosphorus content higher than 100ppm.




```
ggplot(train,aes(P)) +geom_histogram(bins=30, fill="steelblue",color="black")+labs(x="Phosphorus
[ppm]",y="n")+ggtitle("Distribution of Phosphorus")
```

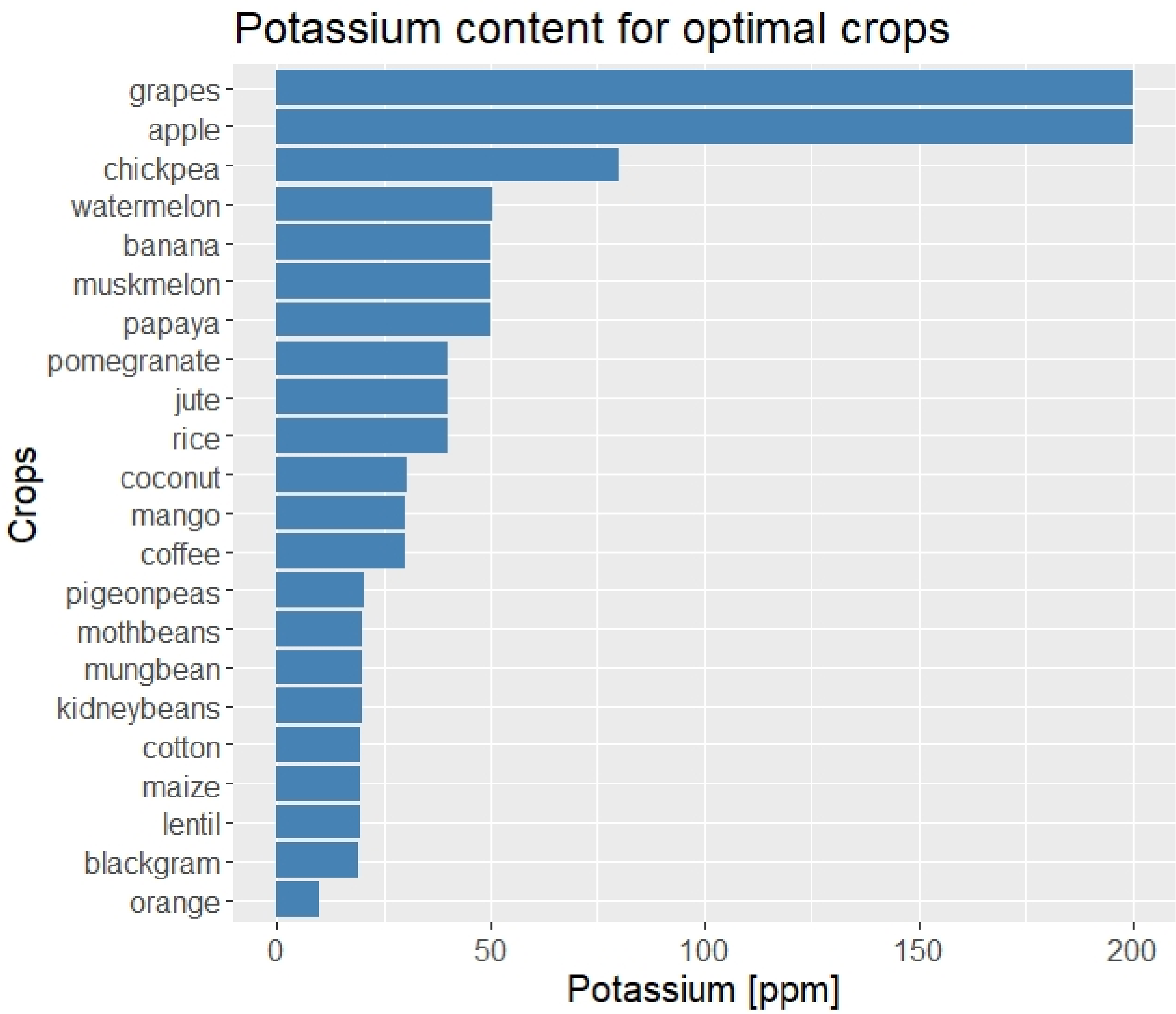



```
ggplot(train,aes(P)) +geom_density()+labs(x="Phosphorus [ppm]",y="Density")+ggtitle("Distributio
n of Phosphorus")
```

Potassium

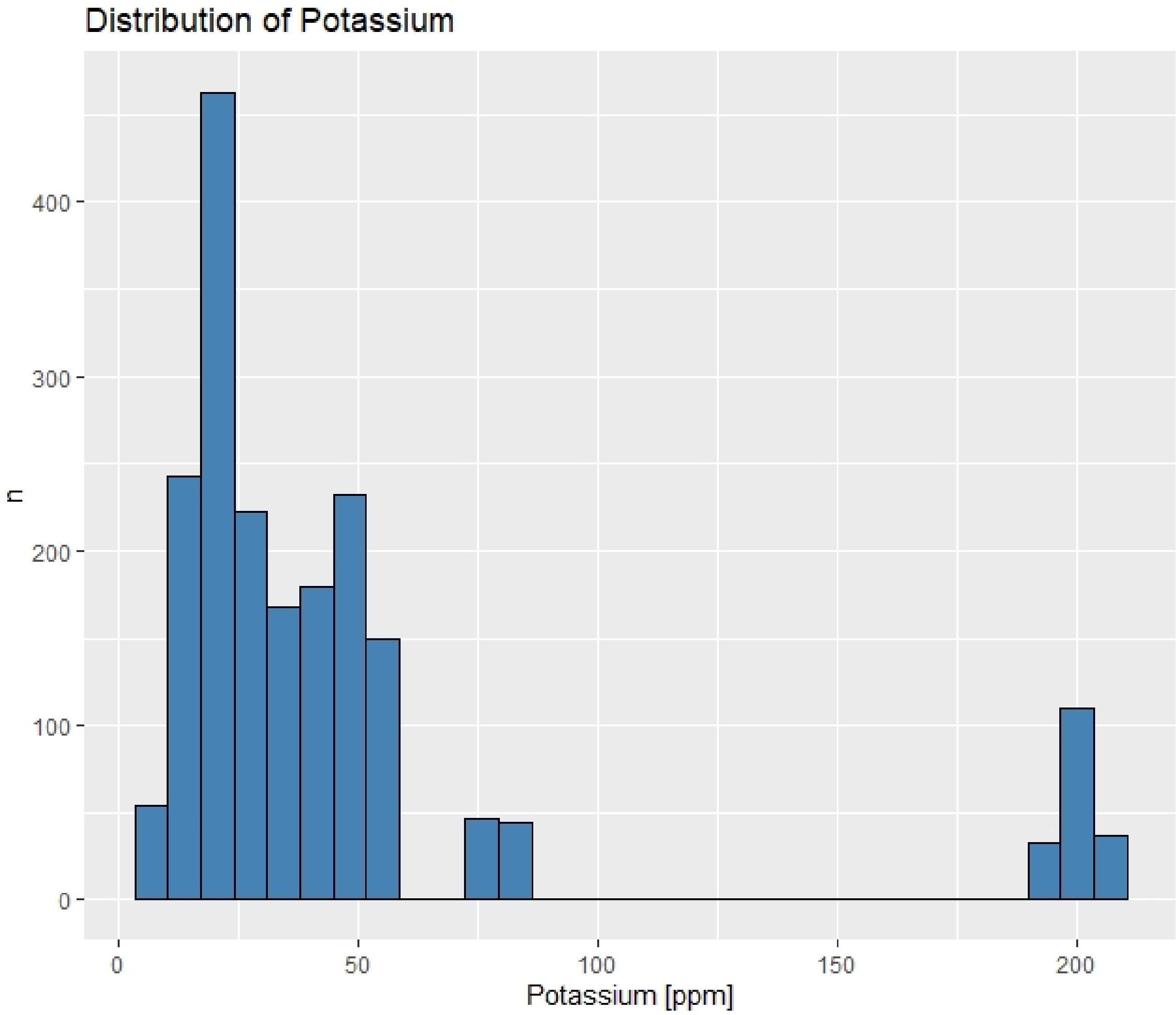
Potassium helps photosynthesis, the process through which the sugars and energy that the plant needs for its development are formed and converted. Potassium also controls the opening and closing of the leaf stomata, which regulate the water status in the plant. Grapes and apples need soil with very high content of potassium, whereby again orange needs a very low potassium content followed by lentil. A list of all mean Potassium values is summarized in the following table and bar chart:

```
train %>%group_by(label)%>%summarise(mean(K)) %>% print.data.frame()

         label   mean(K)
1        apple 200.04444
2       banana  50.04444
3    blackgram  19.20000
4     chickpea  80.02222
5      coconut  30.55556
6       coffee  29.92222
7       cotton  19.57778
8       grapes 200.18889
9         jute  39.93333
10 kidneybeans  19.95556
11      lentil  19.41111
12       maize  19.53333
13       mango  29.92222
14   mothbeans  20.12222
15    mungbean  19.98889
16   muskmelon  50.03333
17      orange  10.01111
18      papaya  49.91111
19  pigeonpeas  20.41111
20 pomegranate  40.12222
21        rice  39.84444
22  watermelon  50.37778
```

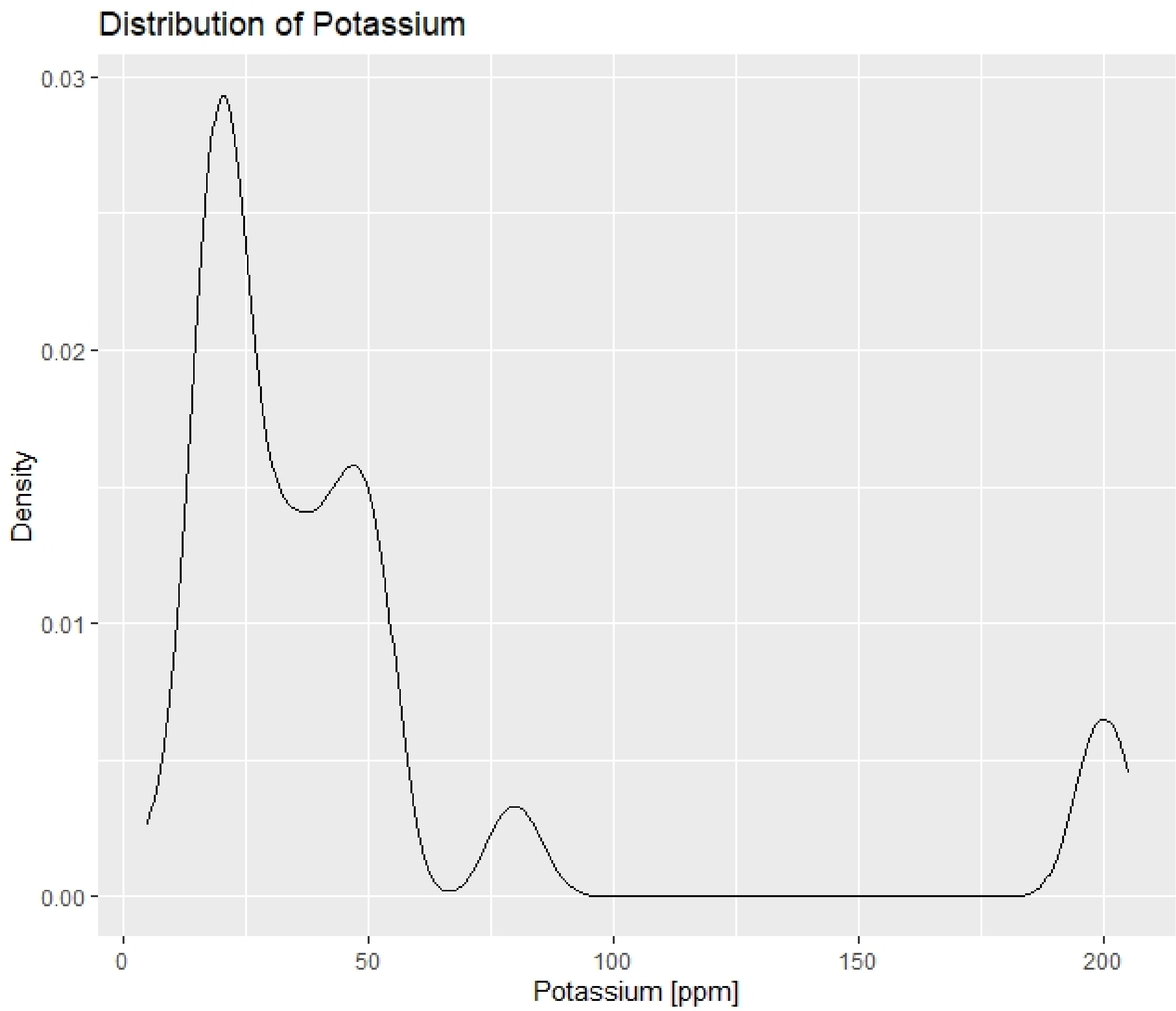


```
train %>% group_by(label) %>% ggplot(aes(x=reorder(label,K/m),y=K/m)) + geom_bar(stat = "identit
y",  fill="steelblue") +labs(x="Crops", y="Potassium [ppm]", title="Potassium content for optima
l crops") +  theme(text = element_text(size=15))+coord_flip()
```

The Histogram and Density chart shows the distribution of Potassium. Just as with phosphorus content, grapes and apple require very high levels of potassium in the soil (around 200ppm). In contrast, all other crops require a potassium content of less than 85ppm.




```
ggplot(train,aes(K)) +geom_histogram(bins=30, fill="steelblue",color="black")+labs(x="Potassium
[ppm]",y="n")+ggtitle("Distribution of Potassium")
```

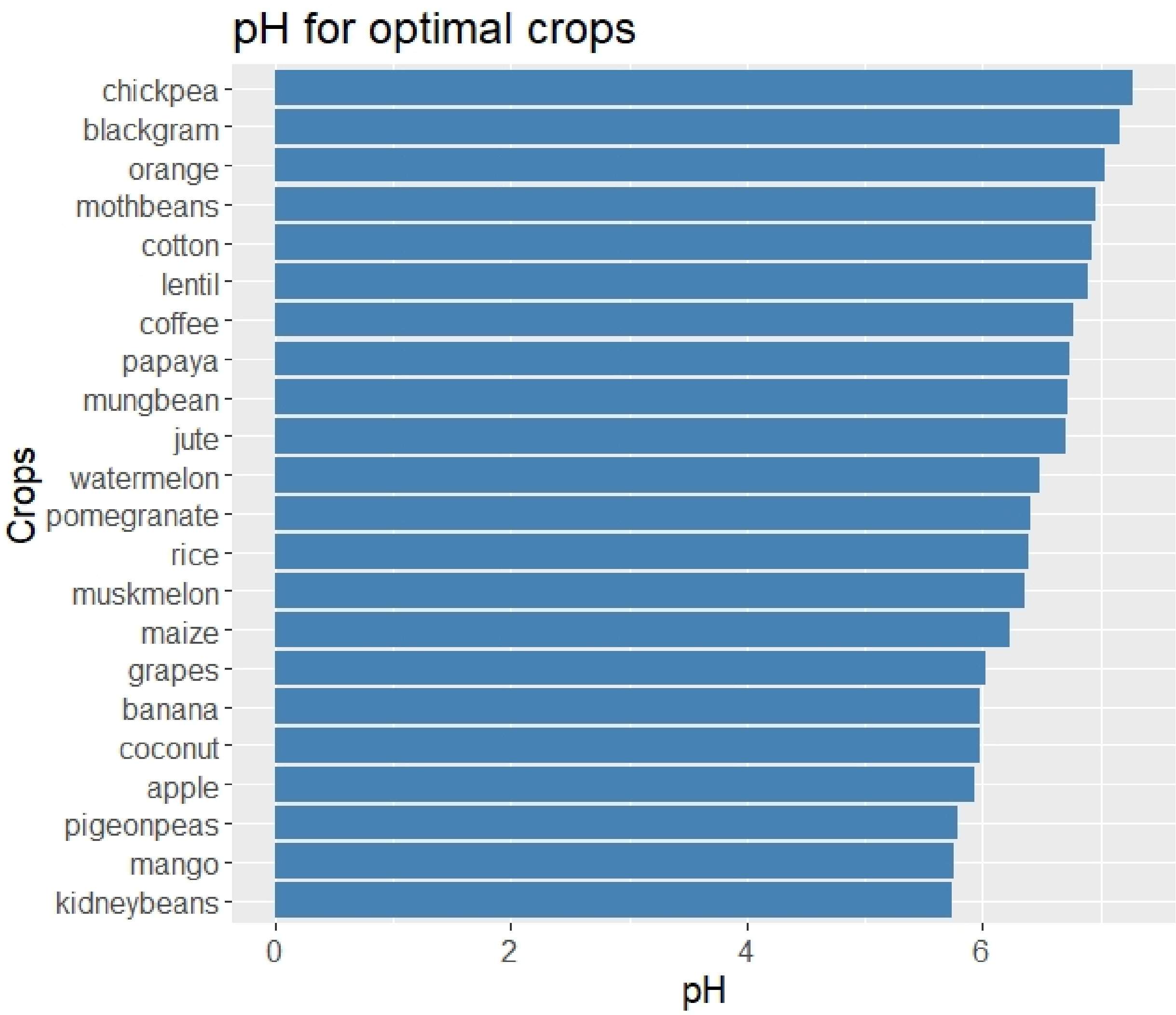



```
ggplot(train,aes(K)) +geom_density()+labs(x="Potassium [ppm]",y="Density")+ggtitle("Distribution of Potassium")
```

pH

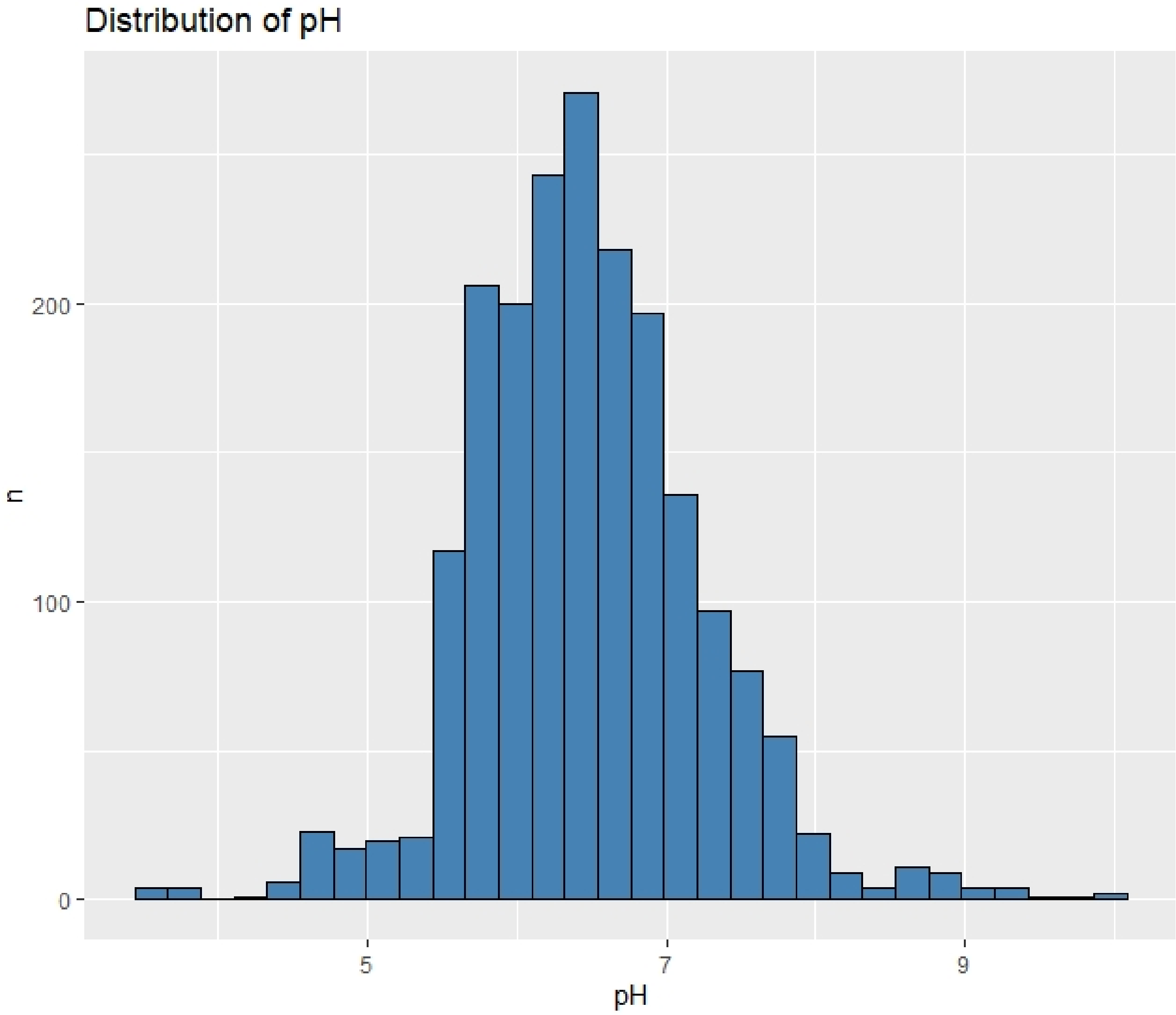
In general, pH values between 6 and 7.5 are optimum for crop and forage production and nutrient uptake. Soil pH impacts nutrient availability and overall soil health. Soil acidification can be an indication of excessive application of nitrogen fertilizer. The range of pH is bigger than the optimal pH range should be in general, and the mean values of the different labels are between 5.74 and 7.27. Chickpea and blackgram need a neutral soil and papaya, mango and kidneybeans need a light acid soil with a pH of about 5.75. A list of all mean pH values is summarized in the following table and bar chart:

```
train %>%group_by(label)%>%summarise(mean(ph)) %>% print.data.frame()

         label mean(ph)
1        apple 5.936633
2       banana 5.980508
3    blackgram 7.159371
4     chickpea 7.278753
5      coconut 5.973727
6       coffee 6.771006
7       cotton 6.924236
8       grapes 6.021094
9         jute 6.714211
10 kidneybeans 5.748075
11      lentil 6.902046
12       maize 6.231523
13       mango 5.764537
14   mothbeans 6.963124
15    mungbean 6.727810
16   muskmelon 6.363348
17      orange 7.038504
18      papaya 6.735134
19  pigeonpeas 5.781993
20 pomegranate 6.413503
21        rice 6.391305
22  watermelon 6.486128
```

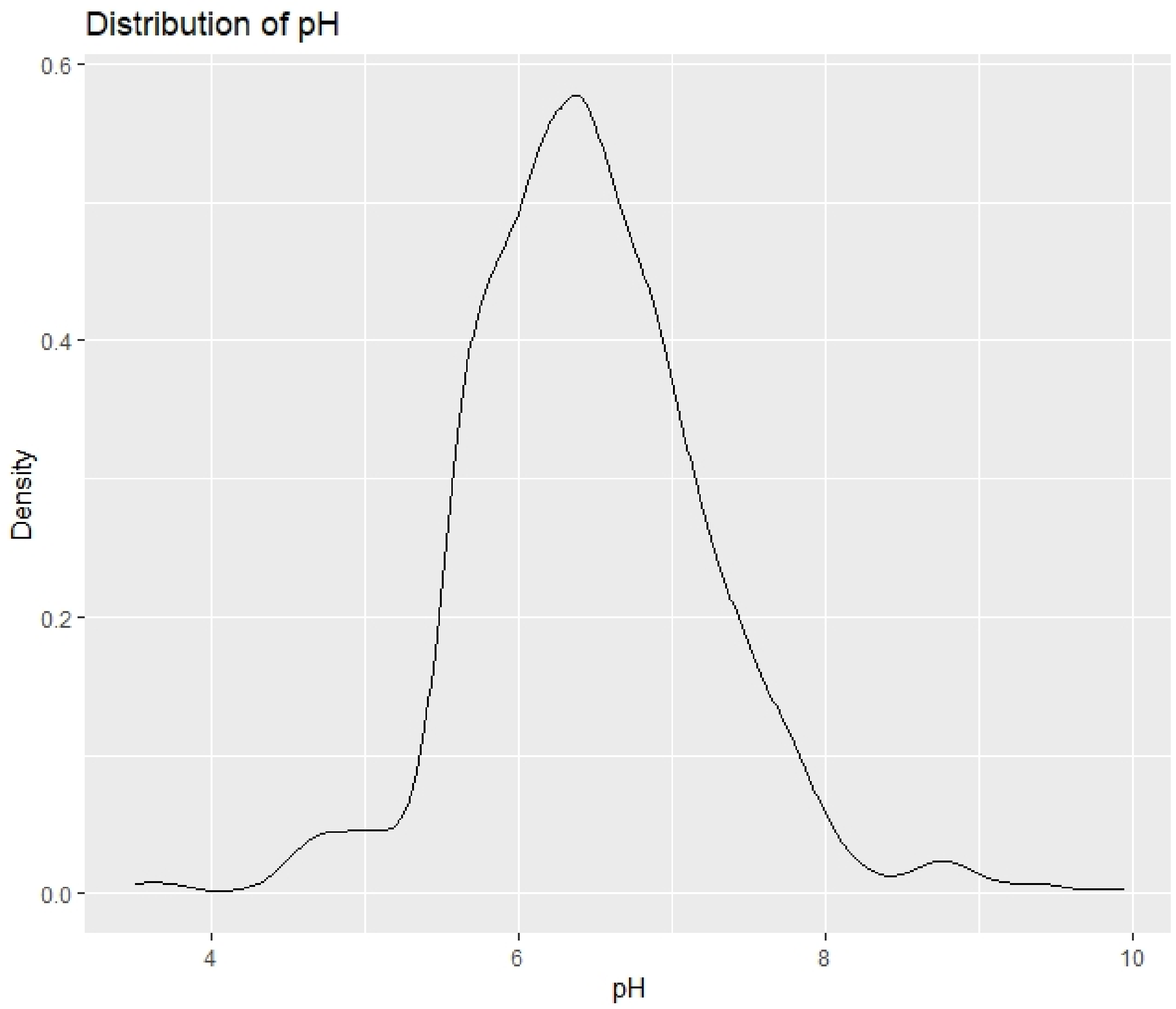


```
train %>% group_by(label) %>% ggplot(aes(x=reorder(label,ph/m),y=ph/m)) + geom_bar(stat = "ident
ity",  fill="steelblue") +labs(x="Crops", y="pH", title="pH for optimal crops") +  theme(text =
element_text(size=15))+coord_flip()
```

The Histogram and Density chart shows the distribution of pH. The pH range is normal distributed with a mean value of 6.47.




```
ggplot(train,aes(ph)) +geom_histogram(bins=30, fill="steelblue",color="black")+labs(x="pH",y
="n")+ggtitle("Distribution of pH")
```

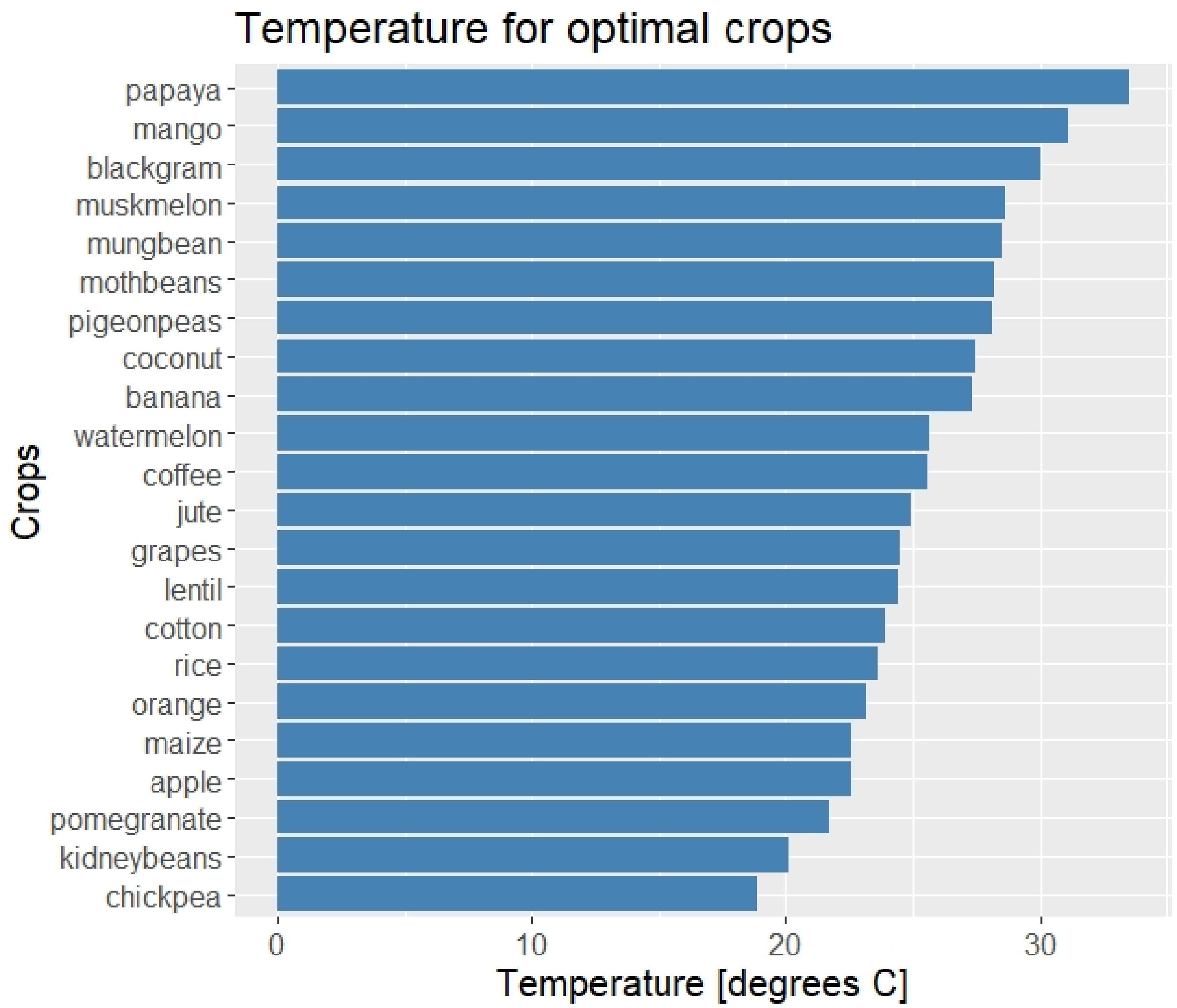



```
ggplot(train,aes(ph)) +geom_density()+labs(x="pH",y="Density")+ggtitle("Distribution of pH")
```

Temperature

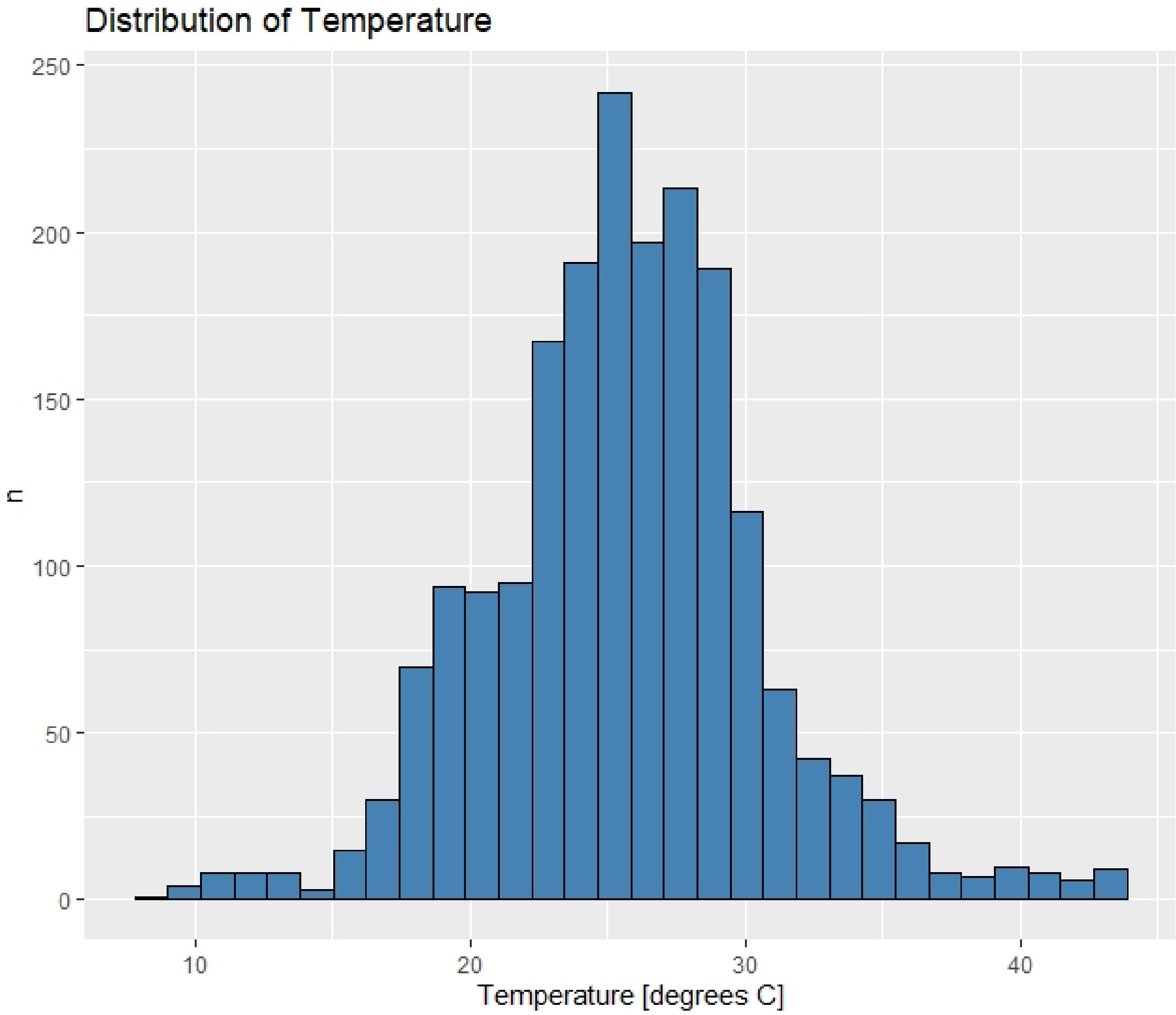
As temperature increases (up to a point), photosynthesis, transpiration, and respiration increase. When combined with day-length, temperature also affects the change from vegetative (leafy) to reproductive (flowering) growth. The range of different temperature is between 18.86 (chickpea) and 30.02 (banana) degrees Celsius. A list of all mean Temperature values is summarized in the following table and bar chart:

```
train %>%group_by(label)%>%summarise(mean(temperature)) %>% print.data.frame()

        label mean(temperature)
1       apple          22.58299
2      banana          27.36189
3   blackgram          30.02464
4    chickpea          18.86139
5     coconut          27.50782
6      coffee          25.61020
7      cotton          23.88927
8      grapes          24.51606
9        jute          24.92283
10 kidneybeans         20.09896
11     lentil          24.45032
12      maize          22.58433
13      mango          31.16318
14  mothbeans          28.18198
15   mungbean          28.53177
16  muskmelon          28.65102
17     orange          23.17387
18     papaya          33.54618
19 pigeonpeas          28.11628
20 pomegranate         21.70613
21       rice          23.63521
22 watermelon          25.63192
```

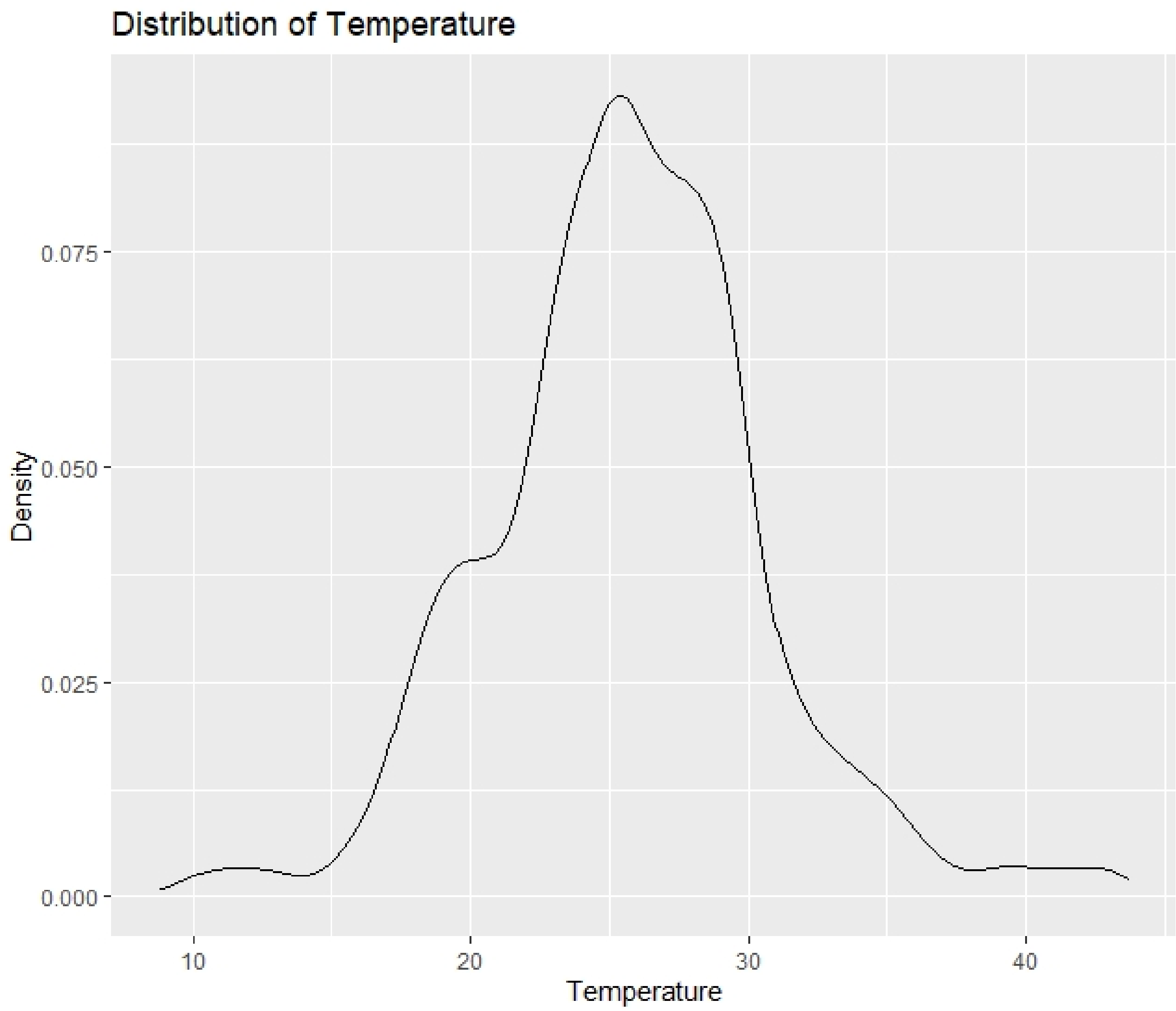


```
train %>% group_by(label) %>% ggplot(aes(x=reorder(label,temperature/m),y=temperature/m)) + geo
m_bar(stat = "identity",  fill="steelblue") +labs(x="Crops", y="Temperature [degrees C]", title
="Temperature for optimal crops") +  theme(text = element_text(size=15))+coord_flip()
```

The Histogram and Density chart shows the distribution of Temperature. Also, the Temperature range is normal distributed with a mean temperature of 25.67 degrees Celsius.




```
ggplot(train,aes(temperature)) +geom_histogram(bins=30, fill="steelblue",color="black")+labs(x
="Temperature [degrees C]",y="n")+ggtitle("Distribution of Temperature")
```




```
ggplot(train,aes(temperature)) +geom_density()+labs(x="Temperature",y="Density")+ggtitle("Distri
bution of Temperature")
```

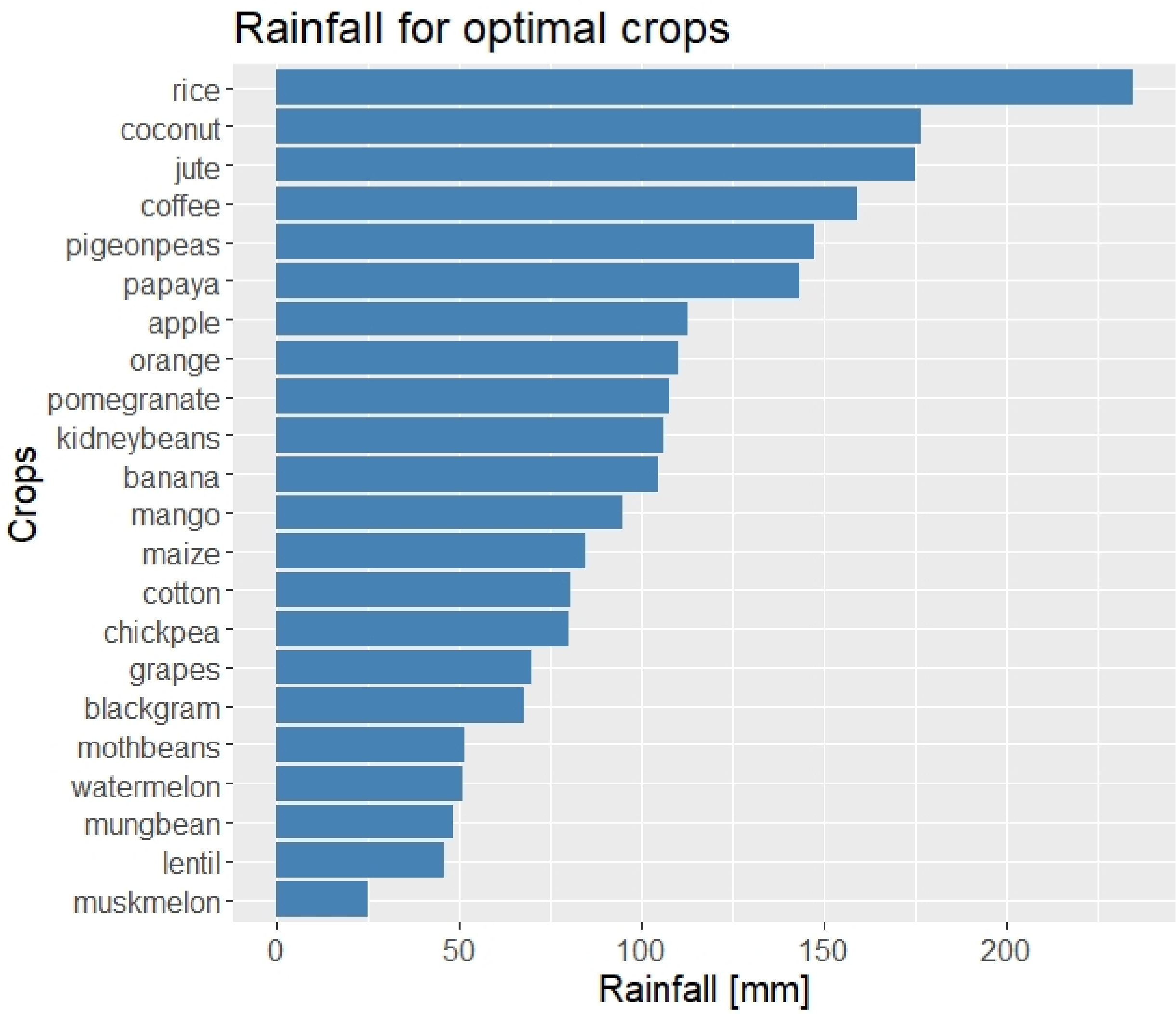
Rainfall

Water helps a plant by transporting important nutrients through the plant. Nutrients are drawn from the soil and used by the plant. Without enough water in the cells, the plant will droop, so water helps a plant to stand upright. Water carries dissolved sugar and other nutrients through the plant.

The water demand of the analyzed plants is very different. Rice, jute and coconut need the most water and muskmelon needs about 10-fold less water than rice. A list of all mean rain amount values is summarized in the following table and bar chart:

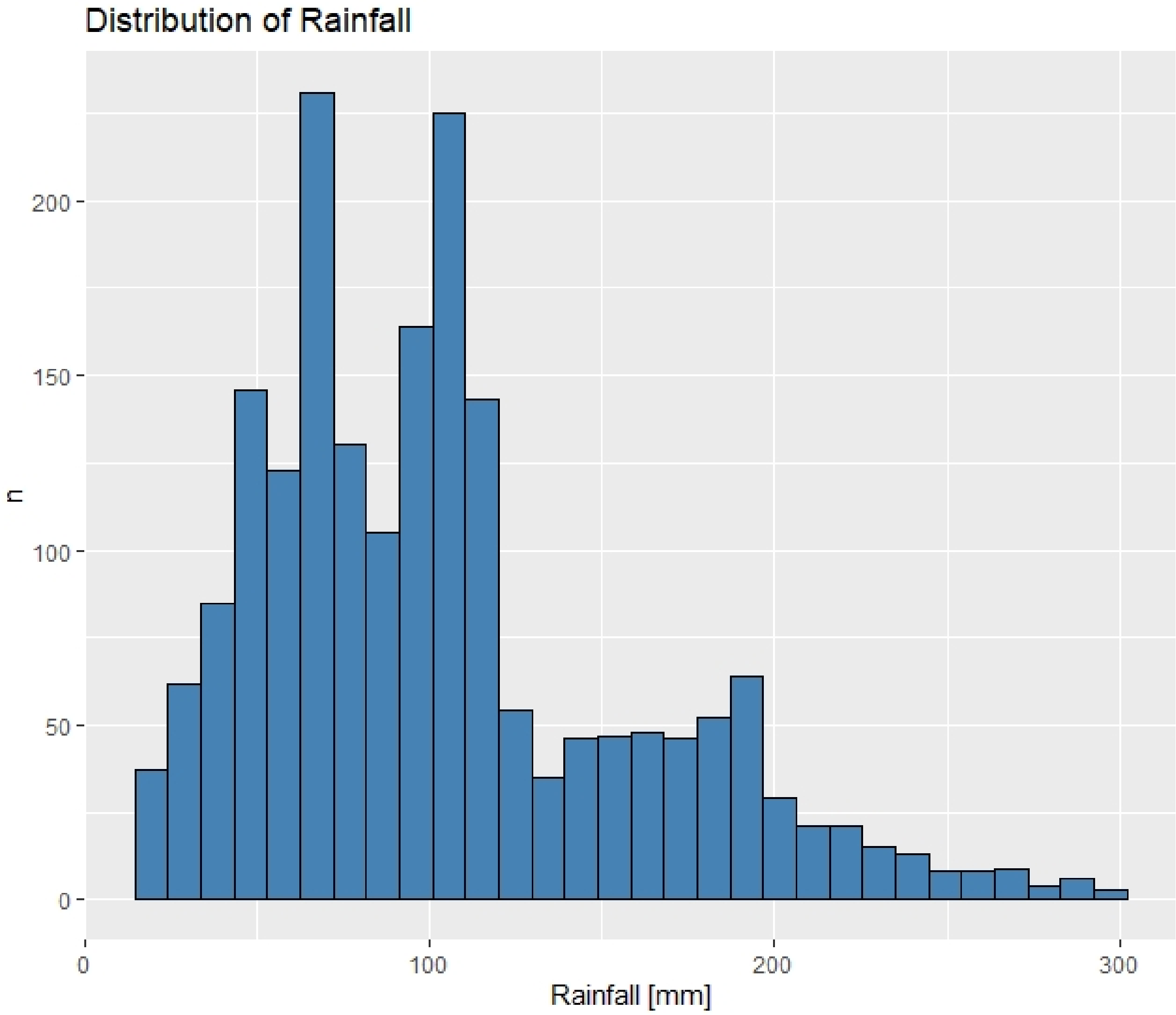
```
train %>%group_by(label)%>% summarise(mean(rainfall)) %>% print.data.frame()

         label mean(rainfall)
1        apple      112.94575
2       banana      104.70680
3    blackgram       67.77173
4     chickpea       79.99625
5      coconut      176.35874
6       coffee      159.14124
7       cotton       80.52756
8       grapes       69.69563
9         jute      174.78475
10 kidneybeans      105.92832
11      lentil       46.06164
12       maize       84.76890
13       mango       94.66307
14   mothbeans       51.55473
15    mungbean       48.27146
16   muskmelon       24.77892
17      orange      110.14881
18      papaya      143.62198
19  pigeonpeas      147.60782
20 pomegranate      107.62403
21        rice      234.83100
22  watermelon       50.98561
```

The histogram shows the different water demands.

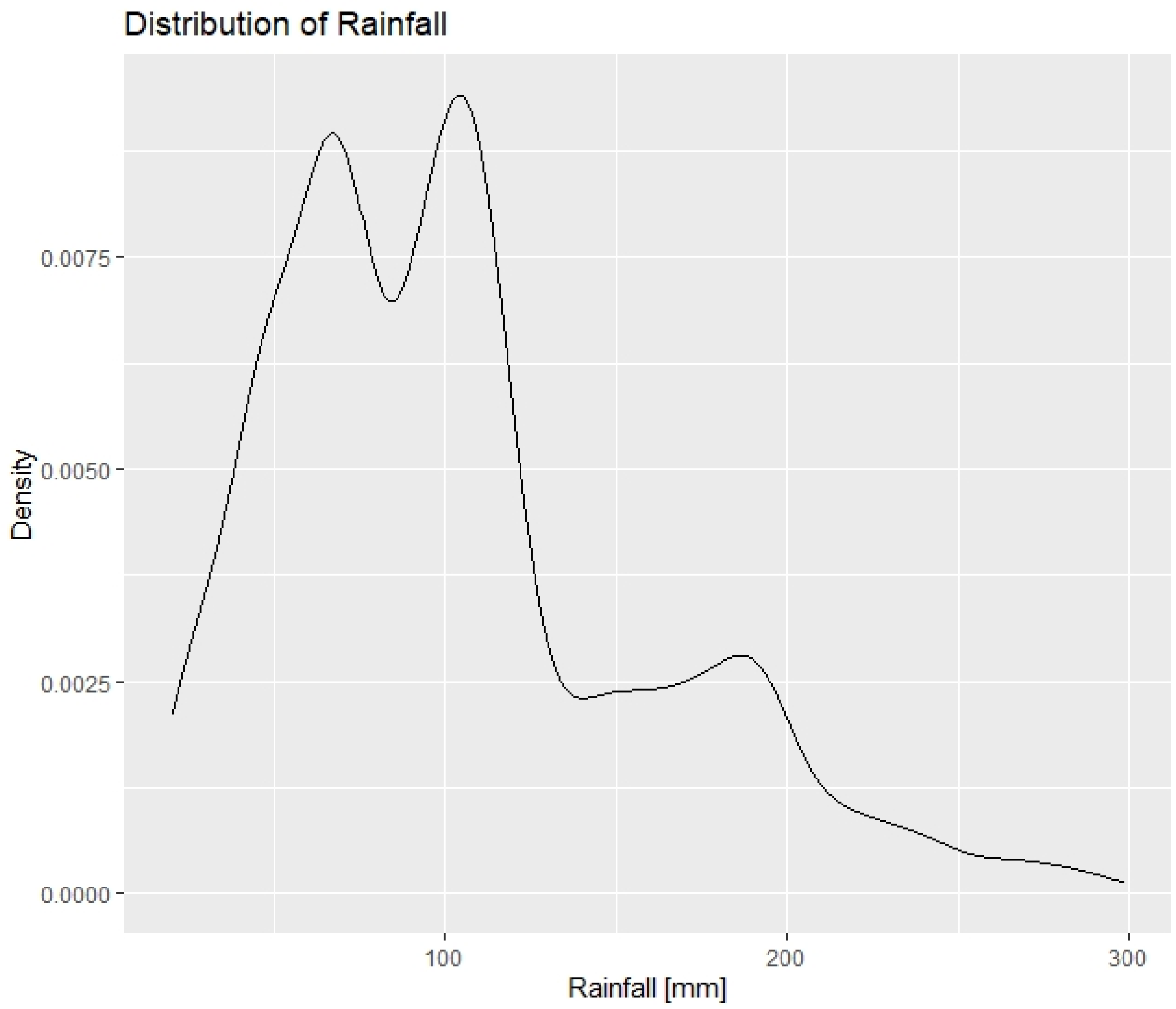


```
train %>% group_by(label) %>% ggplot(aes(x=reorder(label,rainfall/m),y=rainfall/m)) + geom_bar(s
tat = "identity",  fill="steelblue") +labs(x="Crops", y="Rainfall [mm]", title="Rainfall for opt
imal crops") +  theme(text = element_text(size=15))+coord_flip()
```

The Histogram and Density chart shows the distribution of rainfall amount. Individual crops require large amounts of water, but most plants need around 90mm of rainfall for optimal growth.




```
ggplot(train,aes(rainfall)) +geom_histogram(bins=30, fill="steelblue",color="black")+labs(x="Rai
nfall [mm]",y="n")+ggtitle("Distribution of Rainfall")
```

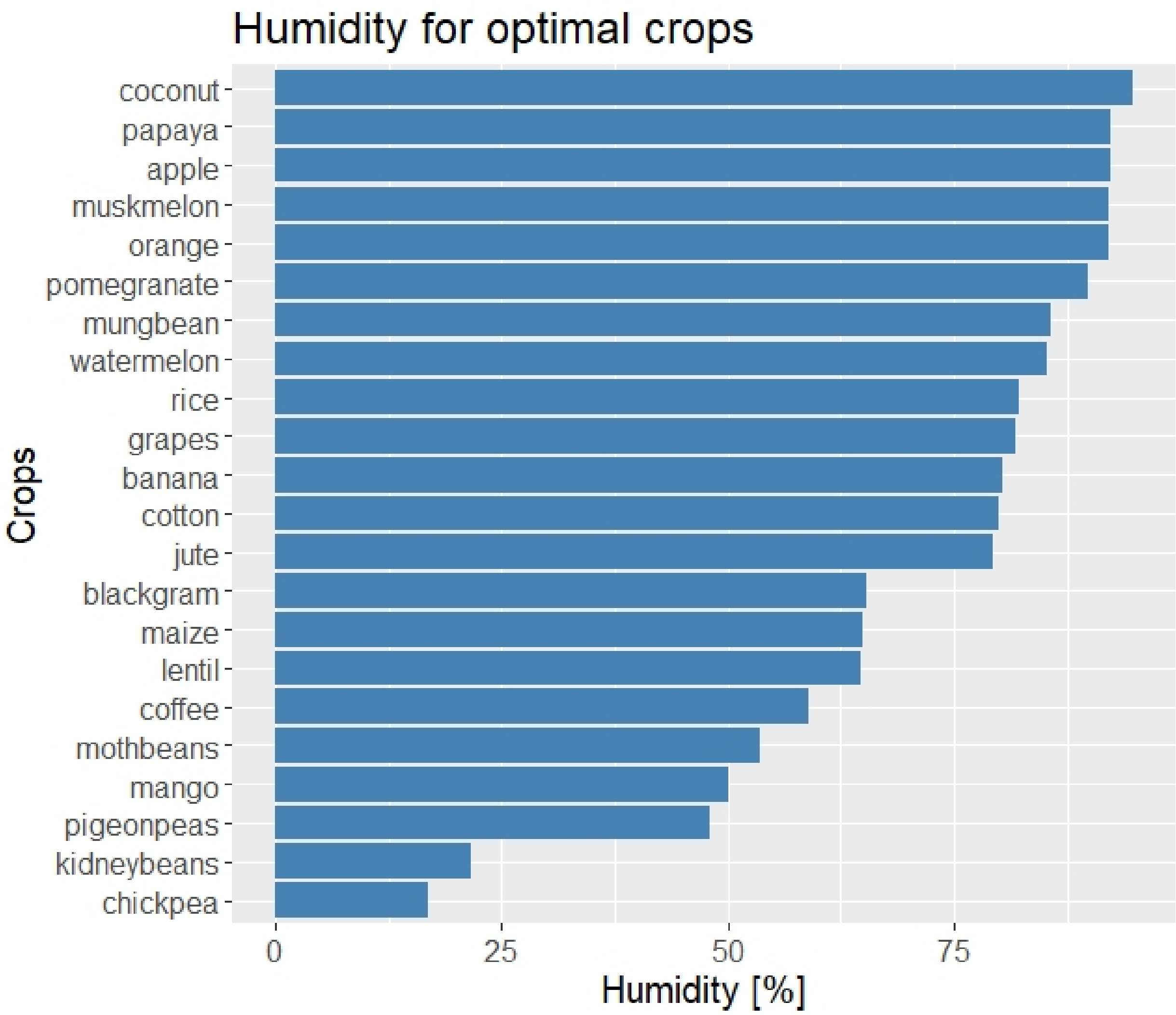



```
ggplot(train,aes(rainfall)) +geom_density()+labs(x="Rainfall [mm]",y="Density")+ggtitle("Distrib
ution of Rainfall")
```

Humidity

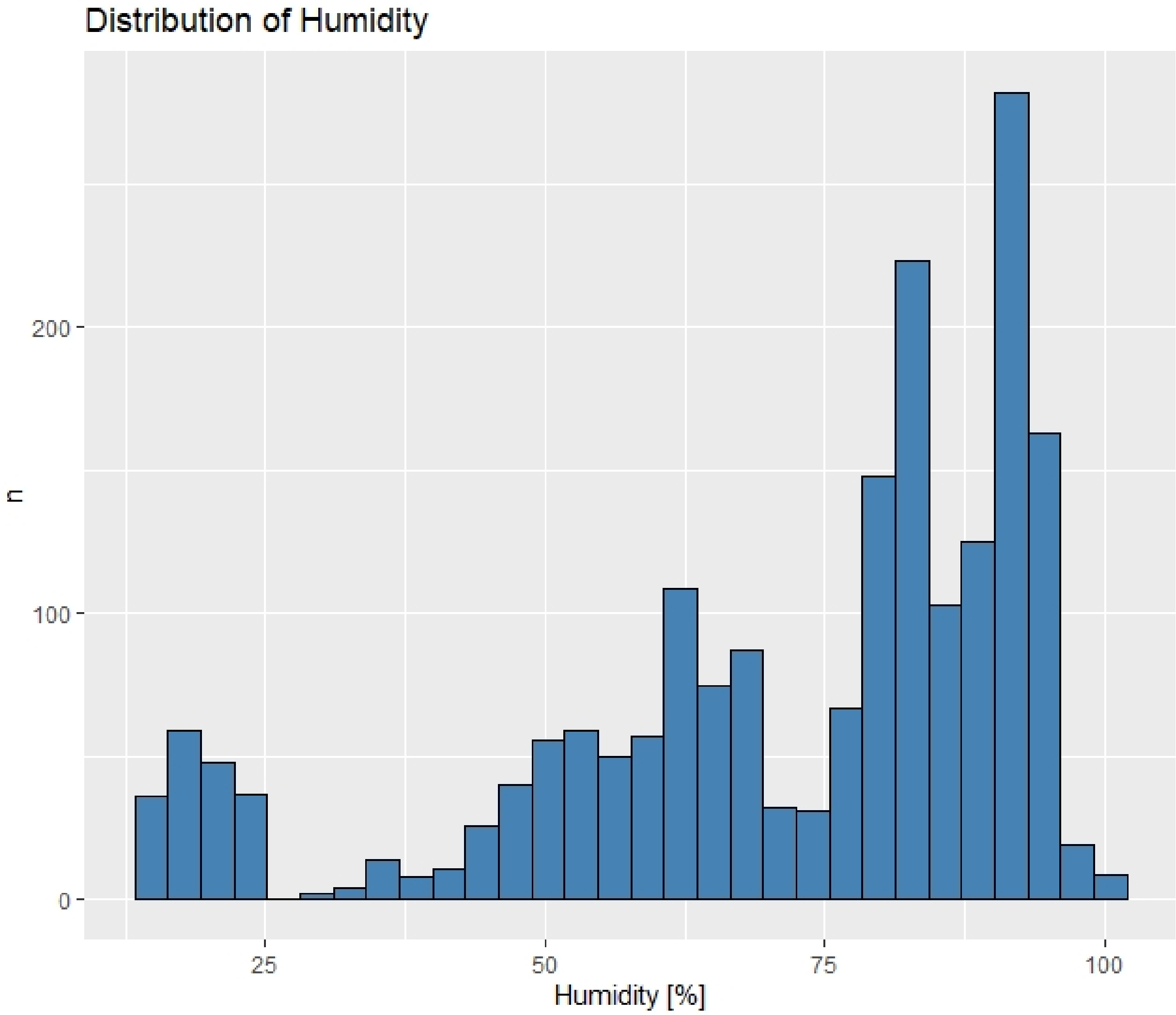
When conditions are too humid, it may promote the growth of mold and bacteria that cause plants to die and crops to fail, as well as conditions like root or crown rot. Humid conditions also invite the presence of pests, such as fungus gnats, whose larva feed on plant roots and thrive in moist soil. Coconut, papaya and apple need the highest humidity and chickpea and kidneybeans need the lowest value. A list of all mean percent of humidity is summarized in the following table and bar chart:

```
train %>%group_by(label)%>%summarise(mean(humidity)) %>% print.data.frame()

         label mean(humidity)
1        apple       92.39294
2       banana       80.42926
3    blackgram       65.34707
4     chickpea       16.89476
5      coconut       94.83994
6       coffee       58.89315
7       cotton       79.96077
8       grapes       81.85577
9         jute       79.35092
10 kidneybeans       21.56208
11      lentil       64.64839
12       maize       64.99231
13       mango       50.08699
14   mothbeans       53.67012
15    mungbean       85.65358
16   muskmelon       92.21952
17      orange       92.21013
18      papaya       92.39890
19  pigeonpeas       48.04678
20 pomegranate       89.92106
21        rice       82.30309
22  watermelon       85.31541
```

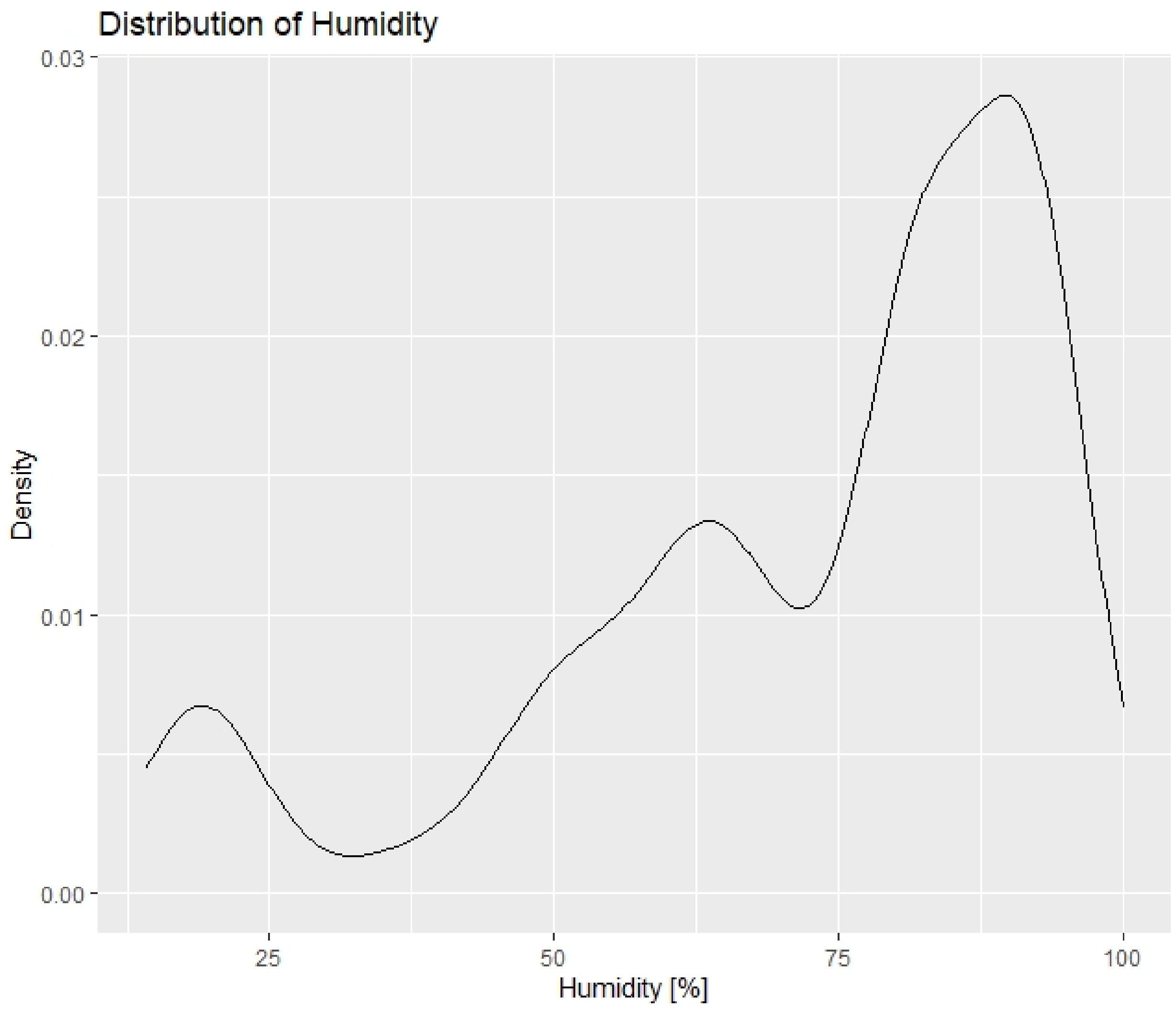


```
train %>% group_by(label) %>% ggplot(aes(x=reorder(label,humidity/m),y=humidity/m)) + geom_bar(s
tat = "identity",  fill="steelblue") +labs(x="Crops", y="Humidity [%]", title="Humidity for opti
mal crops") +  theme(text = element_text(size=15))+coord_flip()
```

The Histogram and Density chart shows the distribution of Humidity. The higher the humidity, the more plants can grow optimally.



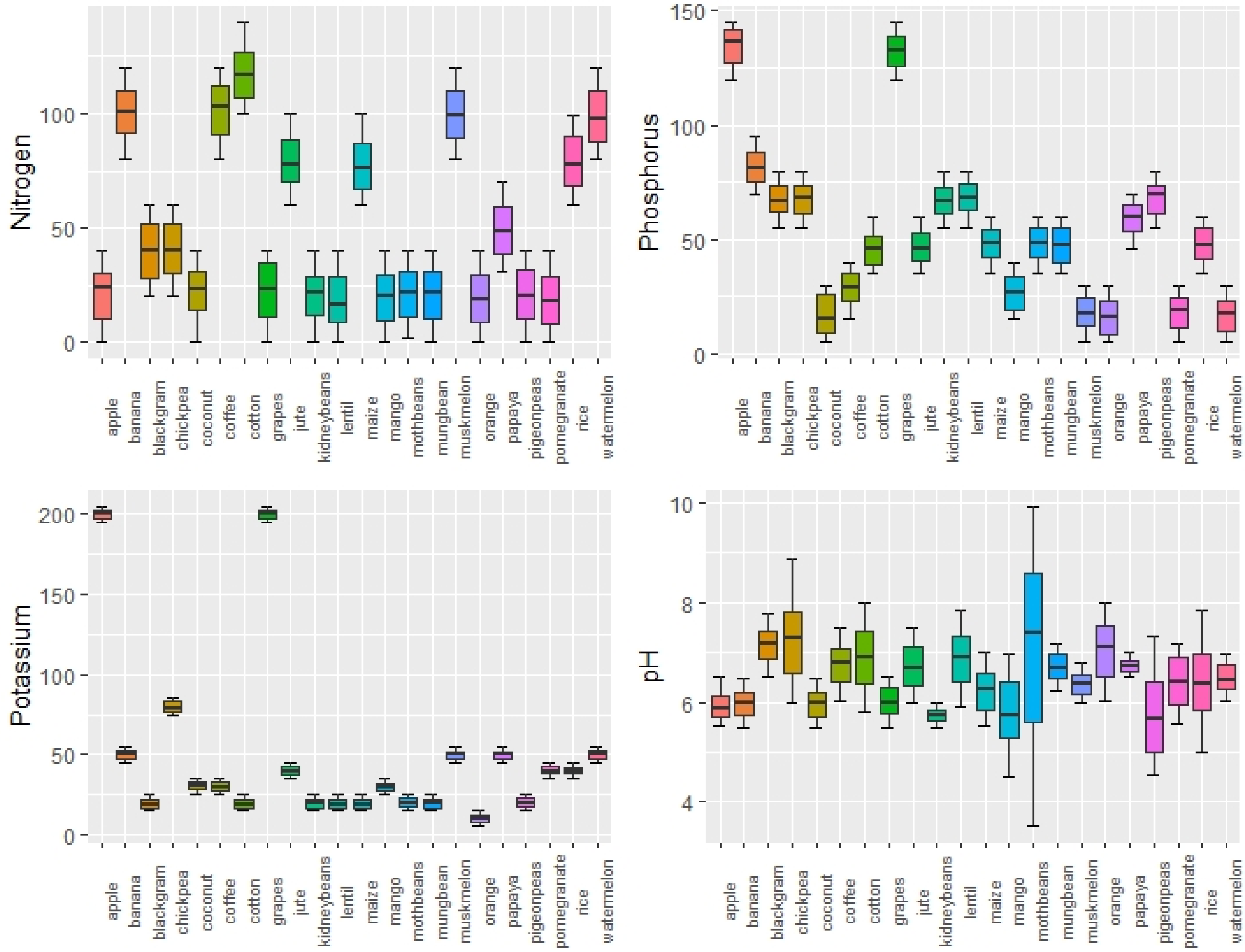

```
ggplot(train,aes(humidity)) +geom_histogram(bins=30, fill="steelblue",color="black")+labs(x="Hum
idity [%]",y="n")+ggtitle("Distribution of Humidity")
```




```
ggplot(train,aes(humidity)) +geom_density()+labs(x="Humidity [%]",y="Density")+ggtitle("Distribu
tion of Humidity")
```

SOIL CONDITION RANGE

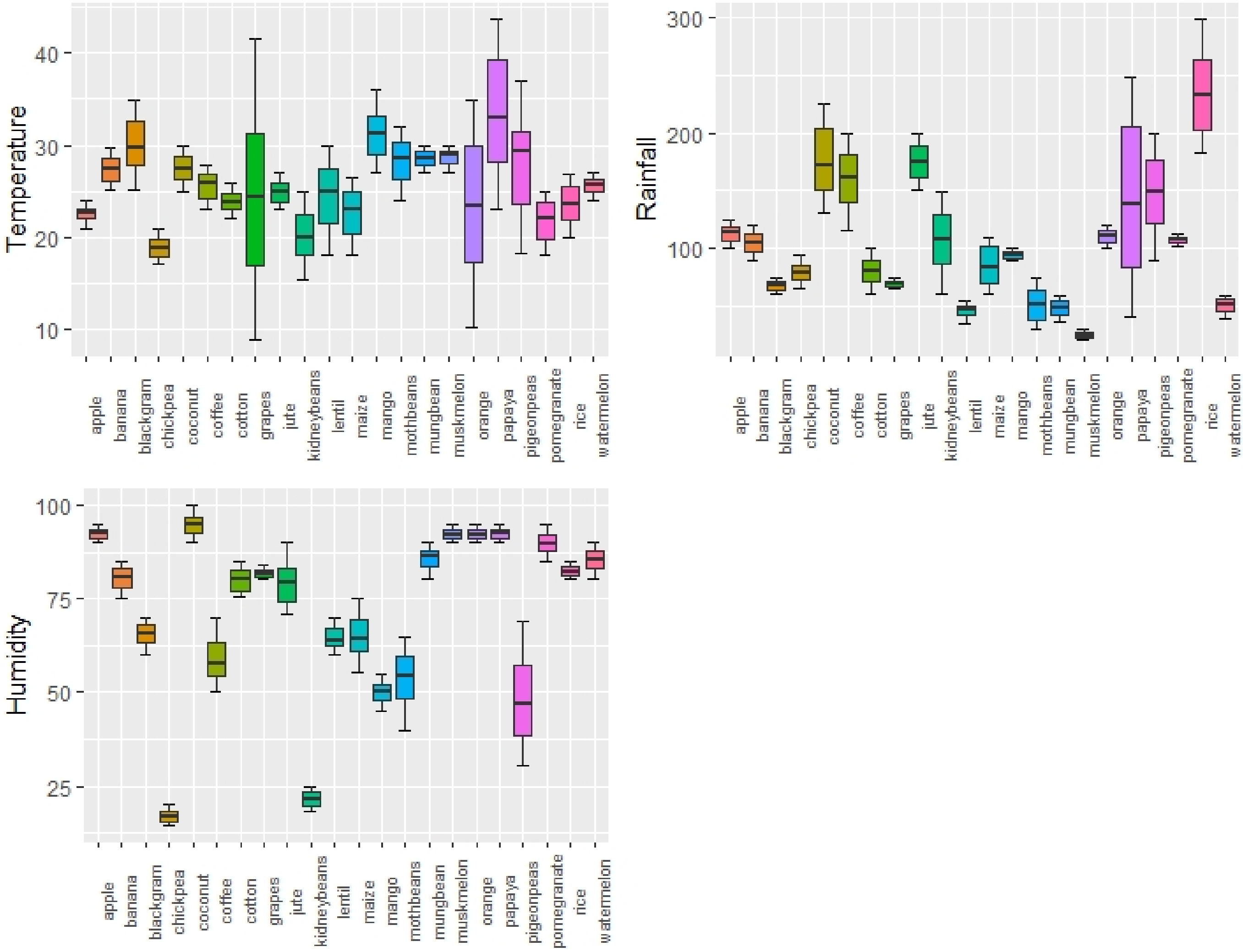
The range of the soul condition Nitrogen, Phosphorus, Potassium and pH are displayed in a following boxplots. For Nitrogen, Phosphorus and Potassium the value ranges are similar between all crops. Compared to the parameters, it is clear that there is only a small range of potassium content for optimal cultivation and harvest. The pH values vary much more among themselves. Very noticeable is the pH range of mothbeans, because it covers the whole range (3.5 - 10).




```
box1 <- ggplot(train,aes(y=N, group=label, x=label, fill=label))+stat_boxplot(geom="errorbar",wi
dth=0.5)+geom_boxplot()+ labs(y="Nitrogen") + theme (axis.text.x=element_text(angle=90, size=7),
legend.position="none", axis.title.x=element_blank())
box2 <- ggplot(train,aes(y=K, group=label, x=label, fill=label))+stat_boxplot(geom="errorbar",wi
dth=0.5)+geom_boxplot()+ labs(y="Potassium") + theme (axis.text.x=element_text(angle=90, size=
7),legend.position="none", axis.title.x=element_blank())
box3 <- ggplot(train,aes(y=P, group=label, x=label, fill=label))+stat_boxplot(geom="errorbar",wi
dth=0.5)+geom_boxplot()+ labs(y="Phosphorus") + theme (axis.text.x=element_text(angle=90, size=
7),legend.position="none", axis.title.x=element_blank())
box4 <- ggplot(train,aes(y=ph, group=label, x=label, fill=label))+stat_boxplot(geom="errorbar",w
idth=0.5)+geom_boxplot()+ labs(y="pH") + theme (axis.text.x=element_text(angle=90, size=7),legen
d.position="none", axis.title.x=element_blank())
grid.arrange(box1,box3,box2,box4, nrow=2)
```

ENVIRONMENT CONDITION RANGE

The ranges of environmental parameters vary much more than the soil parameters.This means that crops with a big parameter range are less sensitive and lead more often to optimal harvest. It also shows that environmental parameters have more influence on the optimal cultivation and harvest of crops. The biggest temperature ranges are for grapes, oranges, papaya and pigeonpeas.That means that these crops are are easier to cultivate. Rainfall sensitivity is lowest for papaya, pigeonpeas, rice and coconut. Humidity sensitivity is lowest for pigeonpeas. In summary, pigeonpeas can be grown in a wide range of environmental parameters and can lead to an optimal harvest.




```
box5 <- ggplot(train,aes(y=temperature, group=label, x=label, fill=label))+stat_boxplot(geom="errorbar",width=0.5)+geom_boxplot()+ labs(y="Temperature") + theme (axis.text.x=element_text(angle=90, size=7),legend.position="none", axis.title.x=element_blank())
box6 <- ggplot(train,aes(y=rainfall, group=label, x=label, fill=label))+stat_boxplot(geom="errorbar",width=0.5)+geom_boxplot()+ labs(y="Rainfall") + theme (axis.text.x=element_text(angle=90, size=7),legend.position="none", axis.title.x=element_blank())
box7 <- ggplot(train,aes(y=humidity, group=label, x=label, fill=label))+stat_boxplot(geom="errorbar",width=0.5)+geom_boxplot()+ labs(y="Humidity") + theme (axis.text.x=element_text(angle=90, size=7),legend.position="none", axis.title.x=element_blank())
grid.arrange(box5,box6,box7, nrow=2)
```

4. RESULTS - MACHINE LEARNING ALGORITHM

To see how this is a type of machine learning, we need to build an algorithm with data we have collected. as users look for crop recommendations. The test set for this calculation of accuracy is provided in the data that will then be applied outside our control, as farmer look for crop recommendations. Lets create a test set to assess the accuracy of the models we implement.

LOSS FUNCTION

We define the rating for movie i by user and donate our prediction. The residual mean squared error RMSE is defined as:

$$RMSE < -function(true_ratings, predicted_ratings)sqrt(mean((true_ratings - predicted_ratings)^2))$$

The lower the RMSE, the better it is.

4.1. FIRST MODEL

We start to build the simplest possible recommendation system. We predict the Nitrogen on the crops regardless of the label. The estimation minimizes the RMSE is the least squares estimation of the rating and is the average of all ratings.

```
mu <- mean(train$temperature)
mu

[1] 25.67037
```

For the prediction of all unknown ratings with Î¼_hat following RMSE will be calculated:

```
naive_rmse <- RMSE(validation$temperature, mu)
naive_rmse

[1] 5.427997
```

The results will be saved in a table.

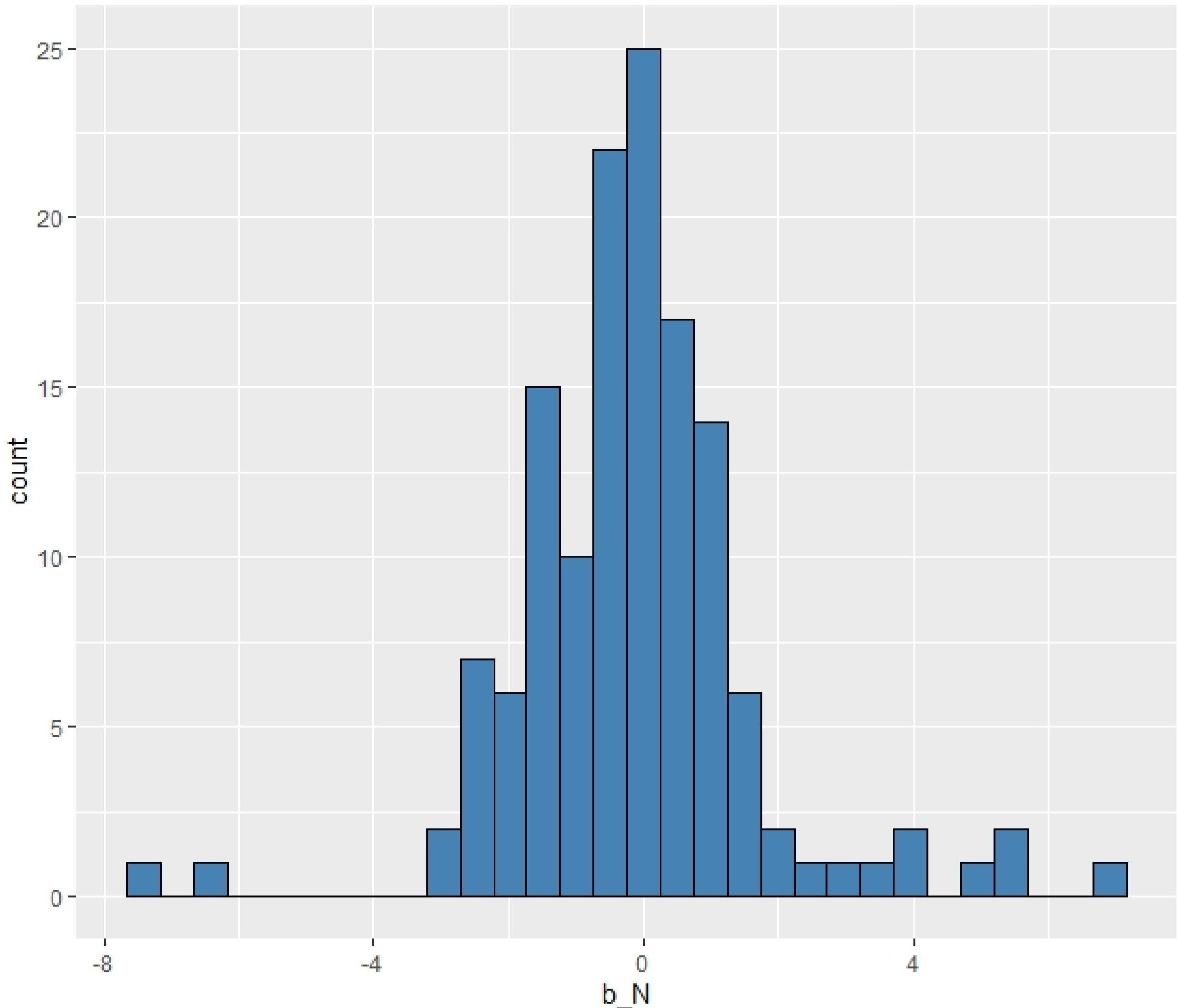
```
rmse_results <- data_frame (method="Just the average", RMSE=naive_rmse)
rmse_results
```

4.2. Nitrogen Effect

The lm() function will be very slow and so I compute it in the whole model using the average like in the following way:

```
mu <- mean(train$temperature)
N_avgs <- train %>% group_by(N) %>% summarise(b_N=mean(temperature-mu))
```

We can see that these estimates vary substantially:




```
train %>% group_by(N) %>% summarise(b_N=mean(temperature-mu)) %>% ggplot(aes(b_N)) +geom_histogr
am(bins=10, color="black", fill="steelblue")
```

We can see how much our prediction improves once we use following calculation: NAs in the data will be changed to Zero, otherwise RMSE could not be calculated.

```
predicted_temperature <- mu + validation %>%left_join(N_avgs, by="N") %>% pull(b_N)
predicted_temperature[is.na(predicted_temperature)]<- 0
N_effect_rmse <- RMSE(validation$temperature,predicted_temperature)
N_effect_rmse

[1] 5.828109
```

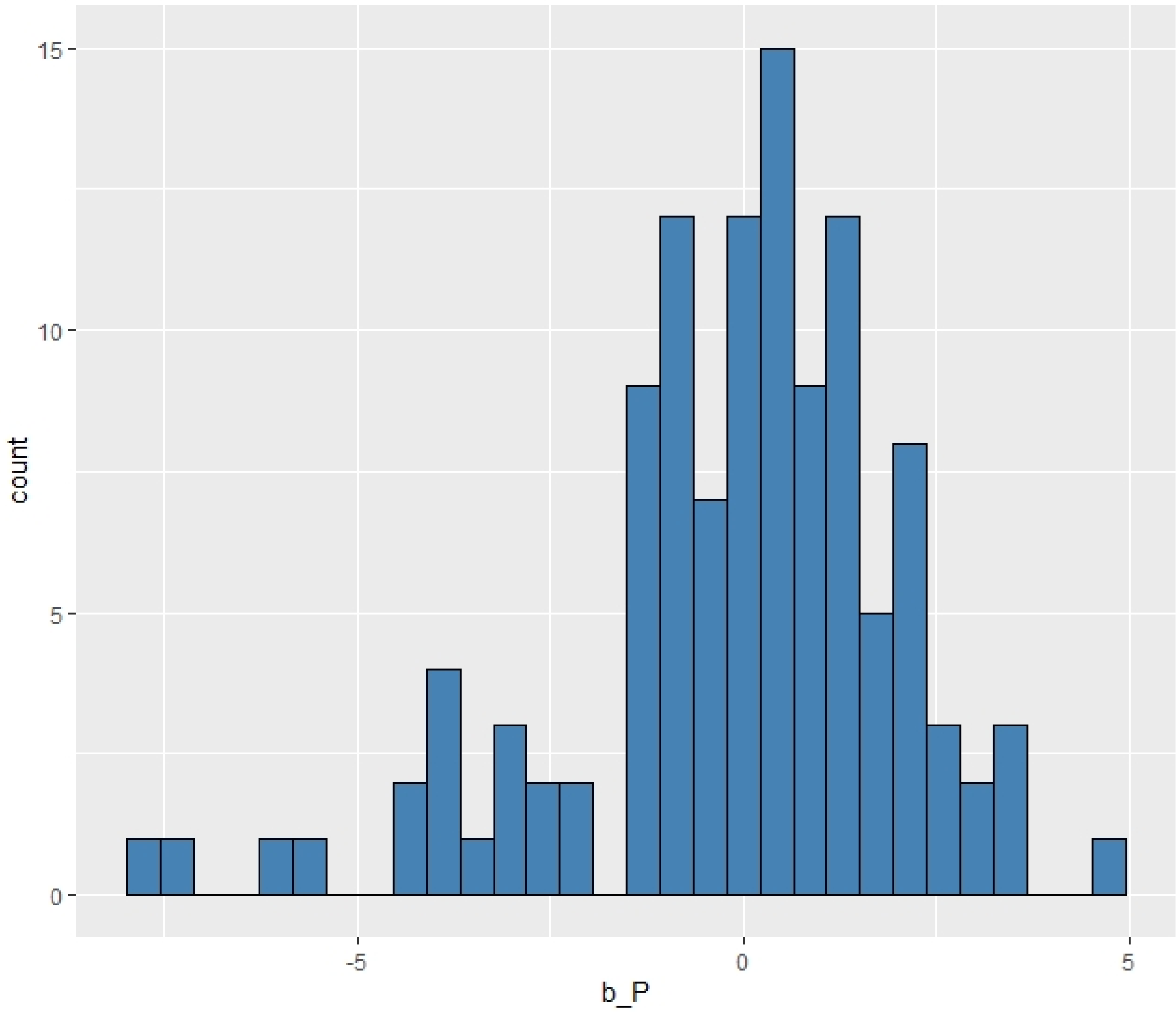
By the Nitrogen Effect the RMSE increase. The results will be saved in the table:

```
rmse_results <- bind_rows(rmse_results, data_frame(method="Nitrogen Effect Model", RMSE=N_effect
_rmse))
rmse_results
```

4.3. PHOSPOURUS EFFECT

The approximation is computed by the phosphorus average, and we can construct predictors and see how much the RMSE improves:

```
mu<- mean(train$temperature)
P_avgs <- train %>% group_by(P) %>% left_join(N_avgs, by="N")%>% summarise(b_P=mean(temperature-
mu-b_N))
```




```
train %>% group_by(P) %>% summarise(b_P=mean(temperature-mu)) %>% ggplot(aes(b_P)) +geom_histogr
am(bins=10, color="black", fill="steelblue")
```

We can compute the predictors and see how the RMSE improve again. NAs in the data will be changed to Zero, otherwise RMSE could not be calculated.

```
predicted_temperature <- validation %>%left_join(N_avgs, by="N") %>% left_join(P_avgs, by="P") %
>%  mutate (pred=mu+b_N +b_P)%>% pull(pred)
predicted_temperature[is.na(predicted_temperature)]<- 0

P_effect_rmse <- RMSE(predicted_temperature,validation$temperature)
P_effect_rmse

[1] 6.006829
```

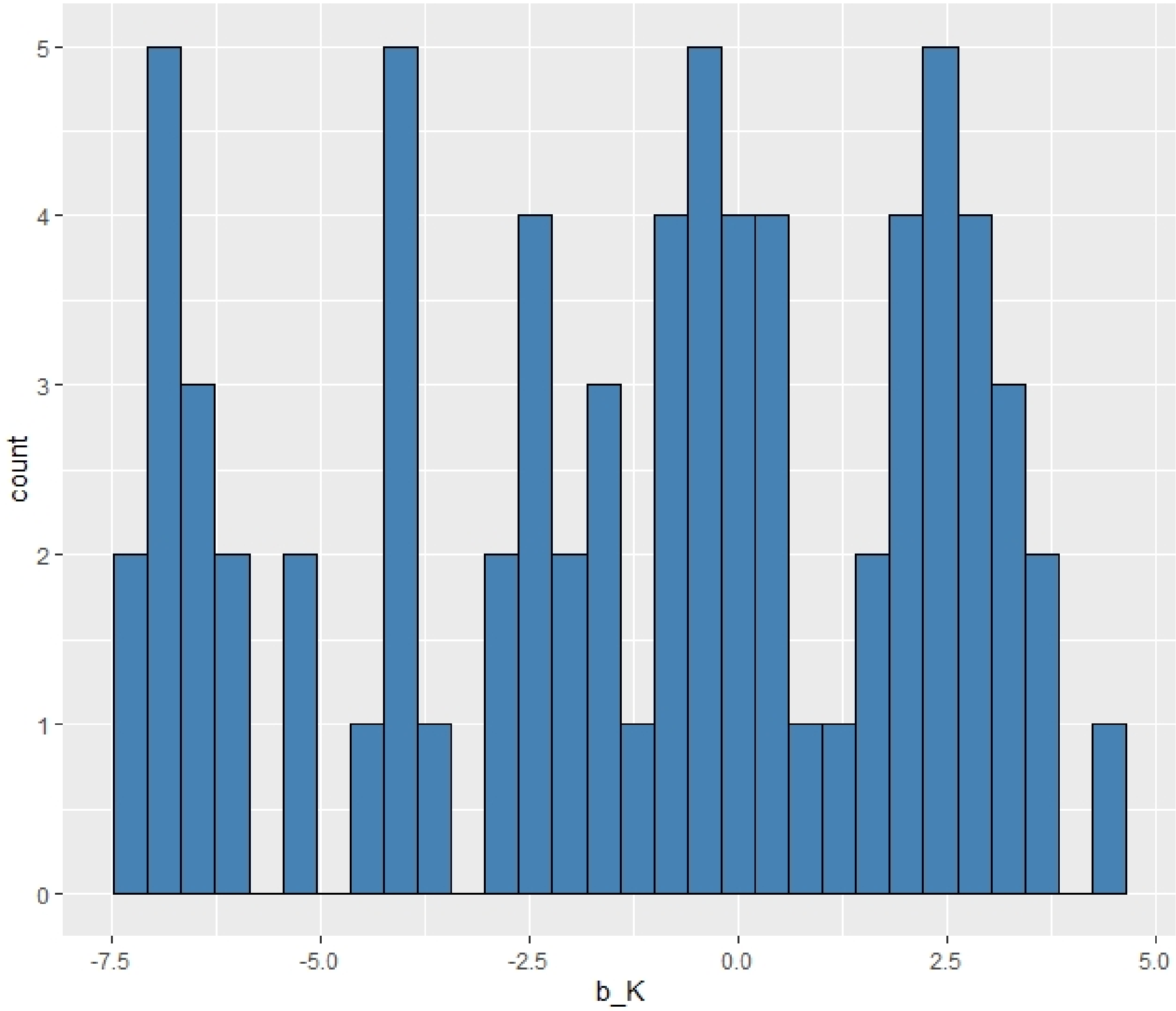
The RMSE increase further. The results will be saved in a table:

```
rmse_results <- bind_rows(rmse_results, data_frame(method="Phosphorus Effect Model", RMSE=P_effe
ct_rmse))
rmse_results
```

4.4. POTASSIUM EFFECT

The approximation is computed by the Potassium average, and we can construct predictors and see how much the RMSE improves:

```
mu<- mean(train$temperature)
K_avgs <- train %>% group_by(K) %>% left_join(N_avgs, by="N") %>% left_join(P_avgs, by="P")%>%
summarise(b_K=mean(temperature-mu-b_N-b_P))
```



```
train %>% group_by (K) %>% summarise(b_K=mean(temperature-mu))%>% ggplot(aes(b_K)) +geom_histogr
am(bins=30, color="black", fill="steelblue")
```

We can compute the predictors and see how the RMSE improve again:

```
predicted_temperature <- validation %>%left_join(K_avgs, by="K") %>% left_join(N_avgs, by="N")%
>%left_join(P_avgs, by="P")  %>%  mutate (pred=mu+b_N +b_P+b_K)%>% pull(pred)
predicted_temperature[is.na(predicted_temperature)]<- 0

K_effect_rmse <- RMSE(predicted_temperature,validation$temperature)
K_effect_rmse

[1] 5.623298
```

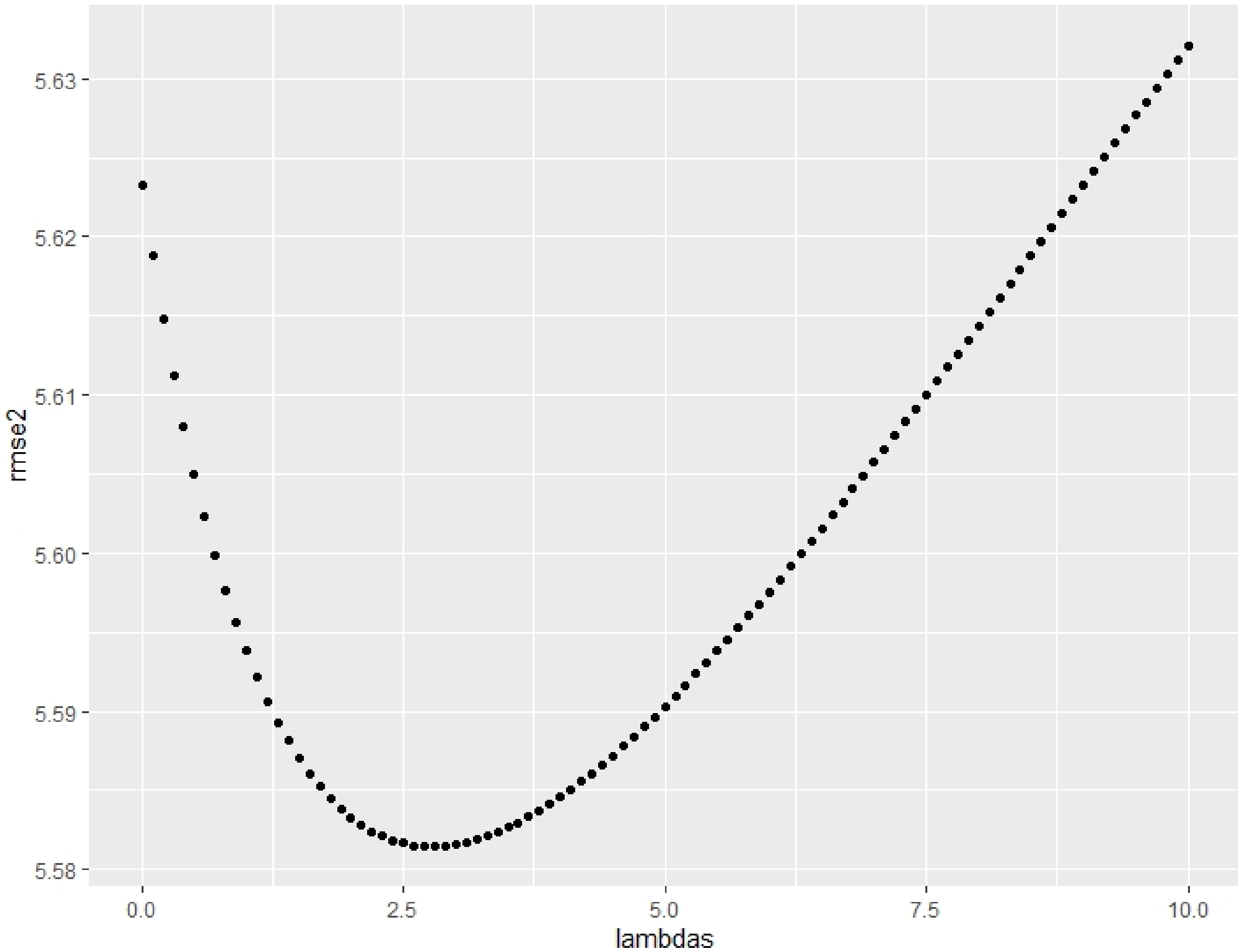
The Potassium Effect let the RMSE decrease. The results will be saved in a table:

```
rmse_results <- bind_rows(rmse_results, data_frame(method="Potassium Effect Model", RMSE=K_effec
t_rmse))
rmse_results
```

4.5. CHOOSING THE PENALTY TERMS

Despite the large variation of the different factors our improvements in RMSE is good. If n is very large, the estimation will be very stable, and the penalty lambda is effectively ignored since ni + lambda is about ni. When the ni is small, then the estimation bi_hat (lambda) is shrunk towards 0. The larger lambda, the more we shrink. We can optimize this model by using lambda. Lets compute these regularized estimate of b_N, b_P and b_K with lambda which is a turning parameter and a sequence between 0 and 10 with distances of 0.25 will be generate and compute the lowest RMSE.

```
lambdas <-seq(0,10,0.1)
mu<- mean(train$temperature)

rmse2 <- sapply(lambdas,function(l){
 mu<-mean(train$temperature)
 b_N_reg<-train %>% group_by(N)%>% summarise(b_N_reg=sum(temperature-mu)/(n()+l))
 b_P_reg<-train %>% group_by(P) %>% left_join(b_N_reg, by="N") %>% summarise(b_P_reg=sum(tempera
ture-mu-b_N_reg)/(n()+l))
 b_K_reg<-train %>% group_by(K) %>% left_join(b_N_reg, by="N") %>% left_join(b_P_reg, by="P") %
>% summarise(b_K_reg=sum(temperature-mu-b_N_reg-b_P_reg)/(n()+l))

 predicted_temperature <- validation %>%  left_join(b_N_reg, by="N")%>%  left_join(b_P_reg,  by
="P")%>% left_join(b_K_reg, by="K")%>%   mutate(pred=mu+b_N_reg+b_P_reg+b_K_reg) %>% pull(pred)
 predicted_temperature[is.na(predicted_temperature)]<- 0
 return(RMSE(predicted_temperature, validation$temperature))})

qplot(lambdas,rmse2)
```




```
lambda_low<- lambdas[which.min(rmse2)]
lambda_low

[1] 2.8
```

The optimal lambda with the lowest RMSE is not calculable. The new RMSE for the model with the optimal lambda is compute as:

```
mu<- mean(train$temperature)
N_reg<-train %>% group_by(N)%>% summarise(N_reg=sum(temperature-mu)/(n()+lambda_low))
P_reg<-train %>% group_by(P) %>% left_join(N_reg, by="N") %>% summarise(P_reg=sum(temperature-mu
-N_reg)/(n()+lambda_low))
K_reg<-train %>% group_by(K) %>% left_join(N_reg, by="N") %>% left_join(P_reg, by="P") %>% summa
rise(K_reg=sum(temperature-mu-N_reg-P_reg)/(n()+lambda_low))

predicted_temperature <- validation %>%  left_join(N_reg, by="N")%>%  left_join(P_reg,  by="P")%
>% left_join(K_reg, by="K")%>%   mutate(pred=mu+N_reg+P_reg+K_reg) %>% pull(pred)
predicted_temperature[is.na(predicted_temperature)]<- 0


All_reg_effect_rmse <- RMSE(predicted_temperature,validation$temperature)
All_reg_effect_rmse

[1] 5.581457

rmse_results <- bind_rows(rmse_results, data_frame(method="Regularized Model with optimal lambd
a", RMSE=All_reg_effect_rmse))
rmse_results


The summarized RMSE are displayed in the following table:

A tibble: 5 x 2
  method                                RMSE
  <chr>                                <dbl>
1 Just the average                      5.43
2 Nitrogen Effect Model                 5.83
3 Phosphorus Effect Model               6.01
4 Potassium Effect Model                5.62
5 Regularized Model with optimal lambda  5.58
```

5. CONCLUSION

The objective of this project was to analyze the crop recommendation data and develop a machine learning algorithm from the dataset “Crop recommendation” from Kaggle website. For this dataset different analyzes were done and a machine learning algorithm was calculated. To see how close the predicted value to the actual value is, the RMSE was calculated. I developed a naive model and different models with effects of rainfall, humidity, Nitrogen, Potassium, Phosphorous and pH with their regarding RMSE. This model is tested with the validation dataset. This algorithm includes an error loss, which is included in the last calculation. The optimal lambda is calculate with 2.8. The end RMSE is calculate with 5.5814and that means that the RMSE is increased by 0.15.