Assignment1-INFO7374

Harshit Shah, Rohit Nikam & Vinaynikhil Hiremath

June 10, 2016

Contents

Introduction

Dataset1: Kiva Dataset

What is Kiva?

An organization that allows people to lend small amounts of money via the Internet to micro- finance organizations in the developing world. Local micro-finance organizations help local business people to post profiles and business plans on Kiva.

Lets see in brief how Kiva works:

1. Kiva loans are facilitated through two models, partner and direct, that enables them to reach the greatest number of people around the world. For partner loans, borrowers apply to a local Field Partner, which manages the loan on the ground. For direct loans, borrowers apply through the Kiva website.
2. The loan then goes through the underwriting and approval process.
3. The loan is posted to Kiva for lenders to support. Depending on the type of loan, a Field Partner or borrower uploads the loan details into the system.
4. Lenders crowdfund the loan in increments of $25 or more.
5. Borrower repays the loan. Lenders receive repayments over time, based on the given repayment schedule and the borrower's ability to repay. The repayments then go into the lenders' Kiva accounts.

The Kiva data set contains a set of heterogeneous information about the following types of entities:

- lenders or kiva users (1,174,383 in total)
- lending teams (25,481 in total)
- loans (564,177 in total)
- field partners (254 in total)
- borrowers (1,099,997 in total)

Dataset2: World Development Indicators Dataset

The primary World Bank collection of development indicators, compiled from officially-recognized international sources. It presents the most current and accurate global development data available, and includes national, regional and global estimates.

World Development Indicators is the World Bank's premier compilation of cross-country comparable data on development. The database contains more than 1,300 time series indicators for 214 economies and more than 30 country groups, with data for many indicators going back more than 50 years. The World Development Indicators offers a condensed presentation of the principal indicators, arranged in their traditional sections, along with regional and topical highlights. The tables include all World Bank member countries (188), and all other economies with populations of more than 30,000 (214 total). Countries and economies are listed alphabetically (except for Hong Kong SAR, China, and Macao SAR, China, which appear after China).

Analysis goals

We are examining the Kiva data set and WDI data set to find the correlation between the food produced and the food imported in a country. We are analyzing the merged data on the food import expenditure of a country and are trying to find the reason for the high and low numbers for different countries.

- To determine the role of agriculture based indicators in the total food imports made by a country.
- To study food produciton indices of a sample of countries and make some inferences.
- To find associative rules between different parameters which will tell us what is affecting the ability of a country to produce food and how it is affecting loans taken by population of a country in agriculture.

Dataset description

We will be working with 2 seperate datasets initially, but ultimately will be merging the 2 datasets into one big dataset on which all out analysis will rely. The 2 datasets are Kiva loans dataset and World Development Indicators dataset.

Kiva Dataset

This data set contains information on 1 million Kiva loans and their repayment structures. Kiva has made its data public so that it can be used for analysis and research.

A list of variables and their definitions are given below:

Variable Name	Data Type	Description
`name`	Character	Name of the person who has taken the loan
`country_code`	Factor	Code assigned to the Country
`country_name`	Factor	Name of the Country
`status`	Factor	Loan status with levels defaulted, in repayment, and paid
`funded_amount`	Numeric	Amount of loan which has been purchased by Kiva lenders
`paid_amount`	Numeric	Amount of the loan which has been paid off
`basket_amount`	Numeric	Amount of the loan which lenders have saved in shopping baskets, but has not been confirmed as purchased
`loan_amount`	Numeric	Payment amount in US dollars
`funded_date`	Date	Date when loan was fully funded on Kiva
`sector`	Factor	Sector for which loan is requested
`activity`	Factor	Activity for which loan is requested
`paid_date`	Date	Date when the loan was repaid
`gender`	Factor	Gender of the borrower

World Development Indicators Dataset

Variable Name	Data Type	Description	Unit
`iso2c`	Factor	ISO Standard 2 character country code	
`country`	Character	Country name	
`year`	Factor	Year	
`iso3c`	Factor	ISO Standard 3 character country code	
`AG.LND.TOTL.K2`	Numeric	Land Area of a country	SQ.KM
`AG.LND.CROP.ZS`	Numeric	Permanent cropland	% of land area
`AG.LND.ARBL.ZS`	Numeric	Arable land	% of land area
`AG.LND.AGRI.ZS`	Numeric	Agricultural land	% of land area
`AG.LND.IRIG.AG.ZS`	Numeric	Agricultural irrigated land	% of agricultural land
`AG.PRD.FOOD.XD`	Numeric	Food Production Index	
`TM.VAL.FOOD.ZS.UN`	Numeric	Food imports	% of merchandise imports
`AG.LND.CREL.HA`	Numeric	Land under cereal production	hectares

Dataset preparation

Below is an overview of the steps taken in order to prepare the data.

Kiva Loans Dataset

First we load the libraries, we believe, we might be needing to load the JSON files and convert them to a data frame.

Load libraries

```
library(dplyr)
library(magrittr)
library(RJSONIO)
library(rlist)
```

The below command `setwd` sets the working directory to the loans folder on our system:

```
  setwd("/Users/Harshit/Desktop/INFO7374 - David Oury/Assignment1/loans/")
```

The next step is to load the JSON files from the loans folder to a data frame in R. For that we start by creating a full path of JSON files so that our next functions can locate the file easily.

```
data.folder <- "/Users/Harshit/Desktop/INFO7374 - David Oury/Assignment1/loans/"
loans.file <- list.files(data.folder, pattern="*.json")
```

The custom function **`dfrow.from.list`** takes a list as input, unlists it, rbinds it and then converts it to a data frame, since working with a data frame is easier than a list.

```
# Function which flattens the list
dfrow.from.list = function(aList) {
  data.frame(rbind(unlist(aList)),
             stringsAsFactors=FALSE)
}
```

Then we wrote another function `readJSONFileIntoDataFrame` which will read all the files and convert them to a big data frame.

```
# Function to convert json into dataframe
readJSONFileIntoDataFrame <-
  function (filename) {
    if(isValidJSON(filename)){
      paste(data.folder,
            filename,
            sep="") %>%
        fromJSON() %>%
        { .$loans } %>%
        list.select(name, country_code = location$country_code,
                    country_name = location$country,
                    activity, sector,
                    funded_amount, paid_amount,
                    loan_amount, gender = borrowers$gender, funded_date, paid_date, status) %>%
        lapply(dfrow.from.list) %>%
        bind_rows()
    }
  }
```

The above function uses many other functions to perform the task. We have explained them below:

`isValidJSON()` - This function checks whether the json file that it is reading is valid or not, meaning is it empty or is the structure of the json file faulty, etc.

`paste()` - This function builds the whole path with filename. eg. `"/Users/Harshit/Desktop/INFO7374 - David Oury/Assignment1/loans/1.json"`

`fromJSON()` - This function reads json from the path we got from `paste` function and parses the data into a list.

`list.select()` - Using this function, we select the columns that we want from each loan to do our analysis. The final data frame will only contain these selected variables, and not all.

`lapply(dfrow.from.list)` - We already saw above what `dfrow.from.list` function does. Here we are using lapply to apply the said function to all the files in the list of json files.

`bind_rows()` - There are 2129 loan files and each file has 500 loans transaction. The `bind_rows()` function will bind each loan row-wise and give a large data frame of all the loans from all the 2129 files.

Now, since there are 2129 files and their size is large (>5gb), we used parallelization in R to load all the files. To use parallelization, we first loaded the library needed to do that.

```
library(parallel)
```

The function `create.loans.df.mc` takes the loans file as input and gives out row binded data frame.

```
create.loans.df.mc = function(loans.file.in) {
  #print(loans.file.in)
  loans.file.in %>% # loans.file.in = lender.file[1:3]
  { mclapply(X=.,
             FUN=readJSONFileIntoDataFrame,
             mc.cores=4) # number of cores to use
  } %>%
    bind_rows()
}
```

Finally, we execute the above function and time it to see how long it takes to run it in parallel.

```
#Function to time the execution
system.time({
  loans.df = create.loans.df.mc(loans.file)
})
```

```
##     user   system  elapsed
## 2127.088   59.007 1103.203
```

Here, we have successfully converted all the kiva loans json files into one dataframe.

WDI Dataset

Now, we load the World Development Indicators dataset using the WDI API. We load the API in R.

```
library(WDI)
```

Then we chose the indicator codes we will be working with for the analysis.

```
indicator.codes.final = c("AG.LND.IRIG.AG.ZS",
"AG.LND.AGRI.ZS",
"AG.PRD.FOOD.XD",
"AG.LND.TOTL.K2",
"AG.LND.CROP.ZS",
"AG.LND.ARBL.ZS",
"AG.LND.CREL.HA",
"TM.VAL.FOOD.ZS.UN",
"SP.POP.TOTL")
```

Then we use the WDI API to load the data we need. `extra=TRUE` gives the extra information like `country_code`, `country_name`, etc.

```
wdi.df <- WDI(indicator=indicator.codes.final, start = 2005, end = 2014,
          extra=TRUE # returns additional variables, see documentation
)
```

Now that we have loaded both the datasets, we will merge them to make a dataframe which has the loans data as well as information about the countries those loans belong to.

We decided to merge the datasets on `country_name` and `year`. Since we do not have any year in the loans data frame, we extract the year from `funded_date` and call it `funded_year`. This will help us merge the two datasets on 2 variables.

Dataset merging

We are deriving the funded year of a loan (`funded_year`) from the funded date (`funded_date`).

```
loans.df$funded_year <- substr(loans.df$funded_date,1,4)
```

Then we convert those variables to factors using which we will be merginf the dataframes.

```
loans.df$funded_year <- factor(loans.df$funded_year)
loans.df$country_name <- factor(loans.df$country_name)

wdi.df$year <- factor(wdi.df$year)
wdi.df$country <- factor(wdi.df$country)
```

Finally we merge the dataframes using the `merge()` function.

```
df.all <- merge(loans.df, wdi.df,
                by.x = c("country_name","funded_year"),
                by.y = c("country","year"))
```

Converting variables to factors and numbers

Now we have one huge dataframe with dimensions of 754504, 29. If we take a look at some of the variables, we notice that variables like `funded_amount` and `loan_amount` are character. Instead they should be numbers. So, our next task is to convert such variables to their respective type.

First, we convert the variables to factors. We create a vector of variable names which we want to convert to factors and we use `lapply` and `factor` function to convert these variables to factors.

```
nms <- c("country_code","activity","sector","status", "funded_year")
df.all[nms] <- lapply(df.all[nms], factor)
```

Then we convert the relevant variables to numeric datatype.

```
df.all$funded_amount <- as.numeric(df.all$funded_amount)
df.all$loan_amount <- as.numeric(df.all$loan_amount)
df.all$paid_amount <- as.numeric(df.all$paid_amount)
```

Renaming variables for better understanding

After merging the Kiva and the WDI dataset we end up with a large data frame with various attributes. We have considered the below mentioned attributes and have renamed them for better understanding.

```
names(df.all) <- c("country_name","funded_year", "name", "country_code",
"activity", "sector", "funded_amount", "loan_amount", "status",
"paid_amount", "funded_date", "paid_date", "iso2c", "AgriIrrigatedLand%",
"AgriculturalLand%", "FoodProdIndex", "LandArea.SQKM", "PermanentCropland%",
"ArableLand%", "LandUnderCerealProd.Hectares", "FoodImports", "TotalPopulation","iso3c", "region",
"capital", "longitude", "latitude", "income", "lending")
```

Variable summaries

We thought it would be better for the reader to see summaries of some of the variables. So, we have listed them below:

`country_code` : This factor variable contains the country code or country name abbreviation. The following list shows country code:

```
##    AF    AL    AM    AZ    BA    BF    BG    BI    BJ    BO
##  2275  1346  5431  8429   608  1043   290  1197  5818 16643
```

`country_name` : This character variable contains the country name of the borrower. The following list shows first five country names in the country list:

```
##            Afghanistan                Albania                Armenia
##                   2275                   1346                   5431
##             Azerbaijan             Bangladesh                 Belize
##                   8429                      0                    187
##                  Benin                Bolivia Bosnia and Herzegovina
##                   5818                  16643                    608
##               Botswana
##                      1
```

`sector` : This character variable represents the sector name for the requested loan. The following list shows sector names:

```
##    Agriculture           Arts       Clothing   Construction      Education
##         168032          15374          48423          12707          16600
##  Entertainment           Food         Health        Housing  Manufacturing
##           1237         188957           6526          26806          10229
##   Personal Use         Retail       Services Transportation      Wholesale
##          11141         166677          56893          23431           1471
```

`loan_amount` : This integer variable represents the amount of money distributed to the borrower in the lender's currency. Summary of the sector is as follows:

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##     25.0    300.0    550.0    793.9   1000.0 100000.0
```

`AG.LND.TOTL.K2` : This is the total land area of country.

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    2830  140000  298200  503200  579300 9388000    5650
```

Visualizations

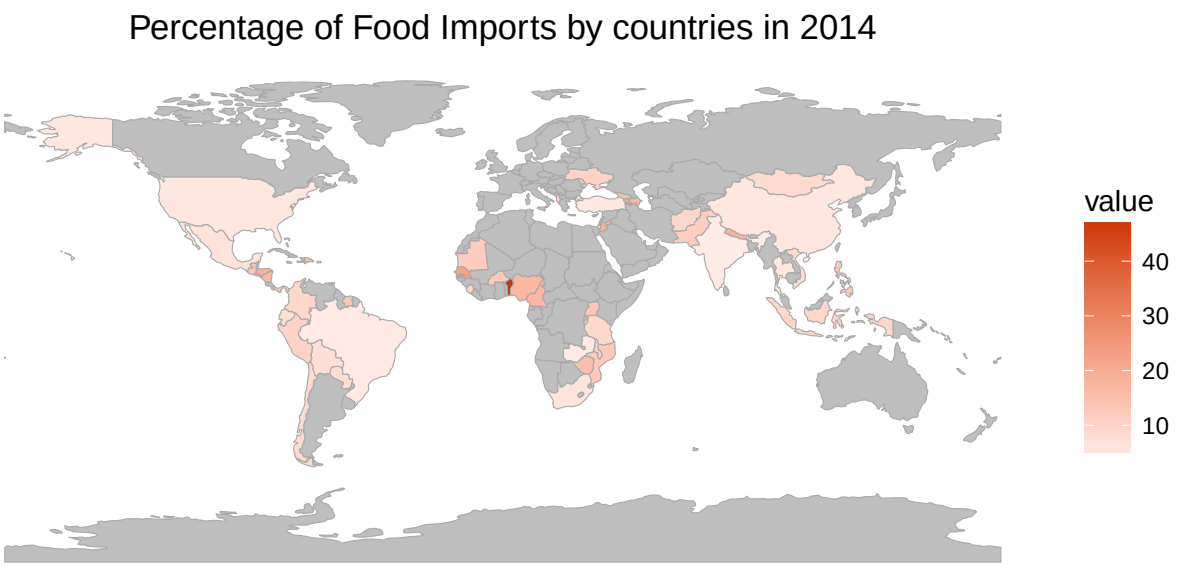
```
library(ggplot2)
library(choroplethr)
library(choroplethrMaps)
```

Geo Plot

In this world map, we are checking the *% of Imports of Food* among total imports of merchandise. For this, we need to first read data from `choroplethrMaps` package which will plot the countries.

```
data(country.regions, package="choroplethrMaps")
```

```
df.all %>%
  select(iso2c, FoodImports, funded_year) %>%
  filter(funded_year==2014) %>%
  unique() %>%
  na.omit() %>%
  merge(country.regions,
        by.x = "iso2c",
        by.y = "iso2c") %>%
  .[,c("FoodImports","region")] %>%
  `colnames<-` (c("value","region")) %>%
  country_choropleth(title = "Percentage of Food Imports by countries in 2014",
                     legend = "%",
                     num_colors=1) +
  scale_fill_continuous(low="#ffebe6", high="#cc3300", na.value="grey")
```



Interpretation: Countries with darker color show that most of their total merchandise import is food, where as countries with lighter shade shows their food import percent is low.

From the above plot we saw that countries like India and United States have lower percent of food imports while Benin has the highest percentage of food imports.

In our analysis, we are taking top and bottom 5 countries based on the %-FoodImport for year 2014. We made 2 different dataframes for top and bottom countries and then binded them together using `rbind()` function.

```
#Most imports
df.all %>%
  select(iso2c, FoodImports, funded_year) %>%
  filter(funded_year==2014) %>%
  unique() %>%
  na.omit() %>%
  merge(country.regions,
        by.x = "iso2c",
        by.y = "iso2c") %>%
  arrange(FoodImports) %>%
  tail(5) -> df.most.5.foodimport

df.most.5.foodimport$level <- (c("high"))

#Least imports
df.all %>%
select(iso2c, FoodImports, funded_year) %>%
  filter(funded_year==2014) %>%
  unique() %>%
  na.omit() %>%
  merge(country.regions,
        by.x = "iso2c",
        by.y = "iso2c") %>%
  arrange(FoodImports) %>%
  head(5) -> df.least.5.foodimport

df.least.5.foodimport$level <- (c("low"))
```

Using `rbind()` function, we will bind these 2 data frames together

```
  df.foodimport.10 <- rbind(df.most.5.foodimport, df.least.5.foodimport)
```

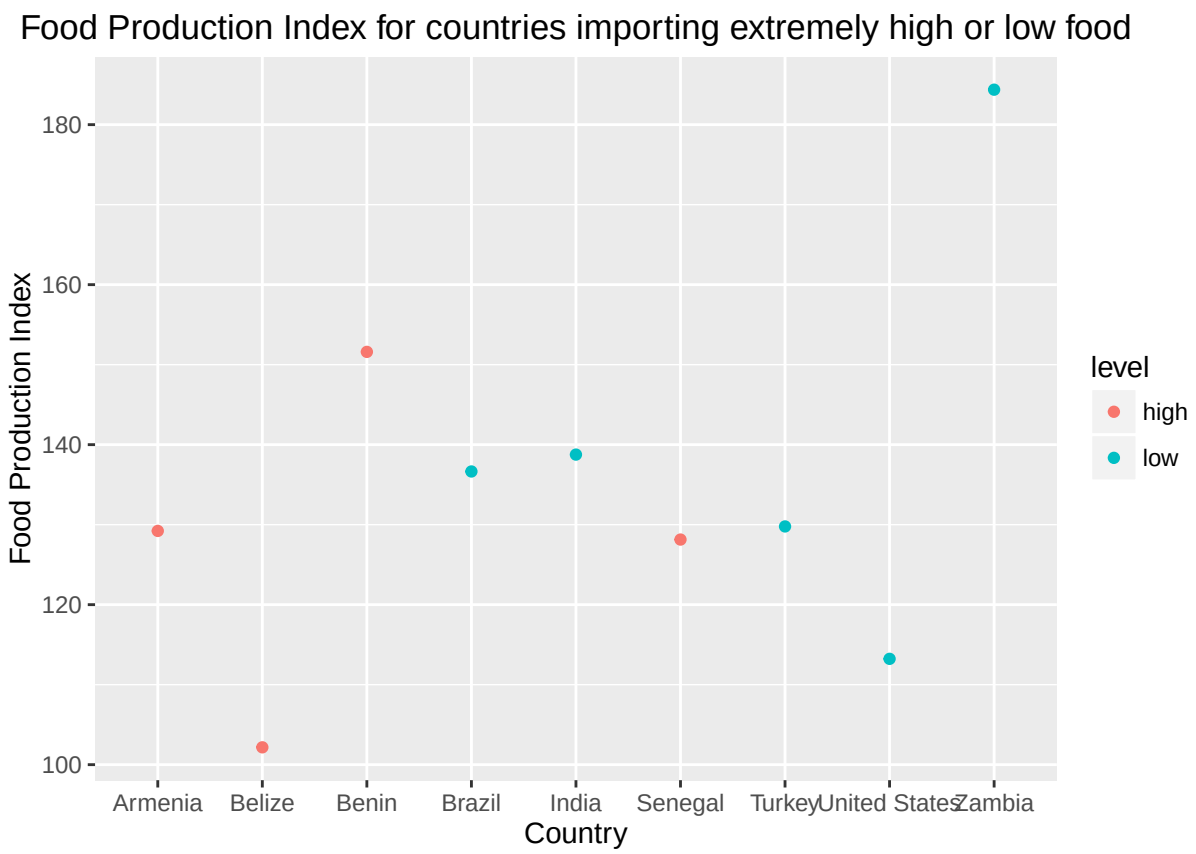
Dot Plot

After we found the countries with minimum and maximum import of foods, we decided to look at Food Production Index for these countries to determine if there is any correlation between Food Production Index and the Food Imports.

We are taking top and bottom 5 countries based on the food imports for year 2014. We made 2 different dataframes for top and bottom countries and then binded them together.

```
df.all %>%
  select(country_name, FoodProdIndex, iso2c, funded_year)%>%
  filter(funded_year==2013) %>%
  unique() %>%
  na.omit() %>%
  merge(df.foodimport.10,
        by.x = "iso2c",
        by.y = "iso2c") %>%
  na.omit() %>%
  arrange(desc(FoodProdIndex)) %>%
  ggplot(aes(country_name, FoodProdIndex)) + geom_point(aes(color = level)) +
  labs(title = "Food Production Index for countries importing extremely high or low food",
```

```
       x="Country",
       y="Food Production Index")
```



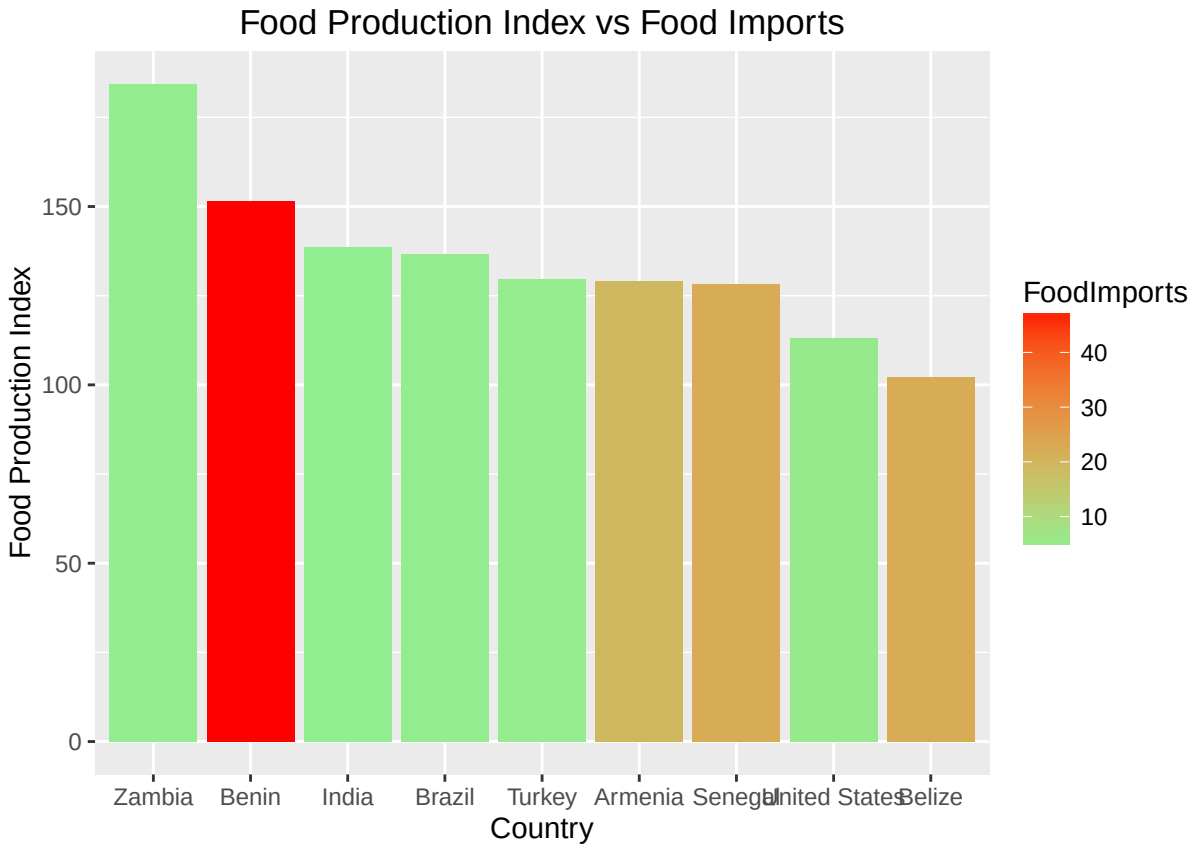
Interpretation: There is no visible correlation between the food production index and food imports of a country. Not all countries support the hypothesis that lower food imports means higher food production index.

Bar Plot

Now we plot Food Production Index and %-FoodImports for the 10 countries to see how they relate to each other.

```
##ggplot bar chart
df.all %>%
  select(country_name, FoodProdIndex, iso2c, funded_year)%>%
  filter(funded_year==2013) %>%
  unique() %>%
  na.omit() %>%
  merge(df.foodimport.10,
        by.x = "iso2c",
        by.y = "iso2c") %>%
  na.omit() %>%
  arrange(desc(FoodProdIndex)) %>%
ggplot(aes(x = reorder(country_name,-FoodProdIndex), y = FoodProdIndex)) +
  geom_bar(stat="identity", aes(fill=FoodImports)) +
  labs(title = "Food Production Index vs Food Imports",
       x="Country",
```

```
      y="Food Production Index") +
  scale_fill_gradient(low = "light green",
                      high = "red")
```



Interpretation: We see in this bar graph that **Benin** being a country which imports most of its food has a very high Food Production Index, infact the second highest. This raises a doubt as to what could be the reason for such a scenario. *We looked up the internet and found that Benin experiences very bad weather and most of its crops are destroyed because of this bad weather.*

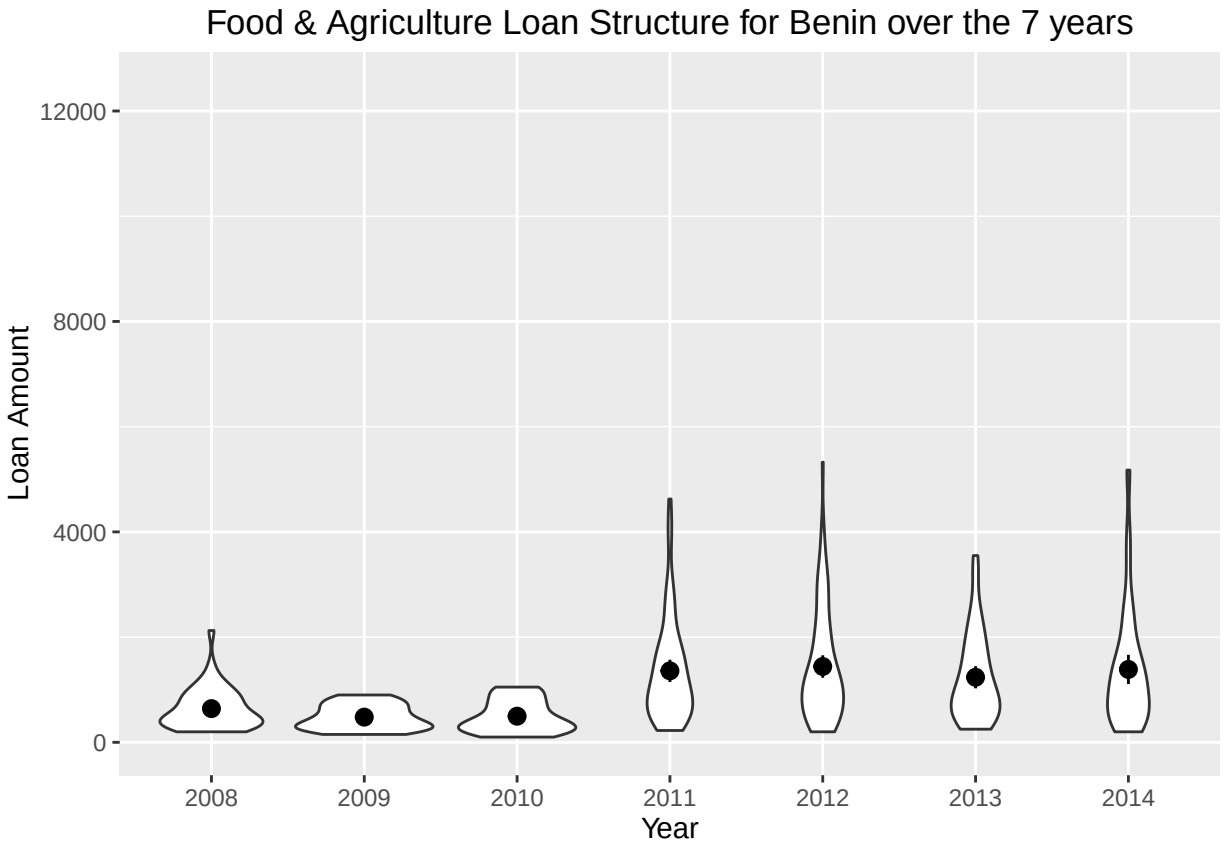
Violin Plot

From the bar plot above we see that Benin clearly has the highest food imports but also has a higher food productions index. To find out the reason behind this aberration we dove deep into the data and chose to analyze the data only related to the country Benin.

Violin plot here is going to help us determine the number and amount of loans for Food and Agricultuer sector for Benin.

```
df.all %>%
  select(sector, loan_amount, funded_year, country_name) %>%
  filter(sector == "Food" | sector == "Agriculture", country_name==c("Benin")) %>%
  unique() %>%
  na.omit() %>%
  ggplot(aes(x=funded_year, y=loan_amount)) +
  geom_violin() +
  stat_summary(fun.data = mean_cl_normal) +
  scale_y_continuous(limits=c(0,12500)) +
  labs(title = "Food & Agriculture Loan Structure for Benin over the 7 years",
```

```
       x="Year",
       y="Loan Amount")
```



Interpretation: Looking deep into Benin's food and agriculture loans, we noticed that historically, people in Benin have takes many loans for food and agriculture. Though the amount of those loans have not been on a higher side, the average loan has nearly doubled over the last 7 years. This could indicate towards many possibilities: Either people are investing in techniques which saves their crop, or people are taking loans to produce more food, or people are just taking loans to buy food.

As Benin had high imports in 2014 despite having high food production index in 2013, we googled to find out why and found that due to erratic weather conditions, a major portion of the produce did not survive.

Association Rules

We find rules which determine the percentage of food imports for a country. For that, we use association rule mining in R.

First we loaded the relavent libraries for association rules.

```
library(arules)
library(arulesViz)
```

Then we decided to look data for 10 years from 2005 till 2014.

```
df.10.yrs <- subset(df.all, funded_year ==
              c("2005","2006","2007","2008","2009","2010","2011","2012","2013","2014"))
```

```
## Warning in is.na(e1) | is.na(e2): longer object length is not a multiple of
## shorter object length
```

```
## Warning in `==.default`(funded_year, c("2005", "2006", "2007", "2008",
## "2009", : longer object length is not a multiple of shorter object length
```

Now, we have some indicators which are % of another indicator. eg. *Arable Land* is a certain percent of *Land Area*. So we calculated these columns to convert them from percentages to definite numbers, in SQ.KM.

For association rule mining, we need to have the variables as factors to be able to find a rule for them. For that, we converted only those variables to factors which we thought would be necessary for determining association rules.

We divide the whole range of loan_amount into quantiles and then rename the values to Q1, Q2, Q3 & Q4, depending in which range the value lies.

First we create factors for loan amount.

```
#create factors for loan_amount
df.10.yrs$loan_amount.factor <- cut(df.10.yrs$loan_amount,
    breaks=
      c(min(df.10.yrs$loan_amount,na.rm=TRUE),
        quantile(df.10.yrs$loan_amount,
                 c(0.25, 0.50, 0.75),
                 na.rm=TRUE),
        max(df.10.yrs$loan_amount,na.rm=TRUE)),
    labels=c('Q1','Q2','Q3','Q4')
)
```

Next we create factors for food imports.

```
df.10.yrs$FoodImport.factor <- cut(df.10.yrs$FoodImports,
   breaks=
      c(min(df.10.yrs$FoodImports,na.rm=TRUE),
        quantile(df.10.yrs$FoodImports,
                 c(0.25, 0.50, 0.75),
                 na.rm=TRUE),
        max(df.10.yrs$FoodImports,na.rm=TRUE)),
   labels=c('Q1','Q2','Q3','Q4')
)
```

Then we convert food production index.

```
#create factors for AG.PRD.FOOD.XD - Food Production Index
df.10.yrs$FoodProdIndex.factor <- cut(df.10.yrs$FoodProdIndex,
   breaks=
     c(min(df.10.yrs$FoodProdIndex,na.rm=TRUE),
       quantile(df.10.yrs$FoodProdIndex,
                c(0.25, 0.50, 0.75),
                na.rm=TRUE),
       max(df.10.yrs$FoodProdIndex,na.rm=TRUE)),
   labels=c('Q1','Q2','Q3','Q4')
)
```

Similarly, we convert land area, permanent crop land, arable land, agriculturable land and agricultural irrigated land to factors.

After we have converted all the required variables to factors, we made a dataframe of all these variables.

```
df.10.yrs %>%
  select(FoodImport.factor, FoodProdIndex.factor, LandArea.factor
         , PermanentCropland.factor, ArableLand.factor,
         AgriculturalLand.factor, AgriIrrigatedLand.factor,
         PopulationPerSqkm.factor, Population.factor) -> df.for.rules
```

Now we have the data frame `df.for.rules` on which we can do association rule mining.

Now we generate rules for food imports.

```
df.rules.fi <-
  apriori(data=df.for.rules,
          parameter=list(maxlen    =4,
                         minlen    =2,
                         support   =0.05,
                         confidence=0.08),
          control = list(verbose=FALSE),
          appearance = list(rhs=c('FoodImport.factor=Q4'),default='lhs'))
```

Then we inspected the rules.

```
inspect(sort(df.rules.fi, by = "support"))
```

```
##    lhs                              rhs                       support confidence     lift
## 1  {LandArea.factor=Q1}          => {FoodImport.factor=Q4} 0.15299590  0.5939799 2.842492
## 2  {Population.factor=Q1}        => {FoodImport.factor=Q4} 0.13271838  0.5308524 2.540395
## 3  {LandArea.factor=Q1,
##     Population.factor=Q1}        => {FoodImport.factor=Q4} 0.12037957  0.5788300 2.769992
## 4  {PermanentCropland.factor=Q3} => {FoodImport.factor=Q4} 0.10774919  0.5538147 2.650281
## 5  {LandArea.factor=Q1,
##     PermanentCropland.factor=Q3} => {FoodImport.factor=Q4} 0.09421759  0.7217259 3.453820
## 6  {AgriculturalLand.factor=Q4}  => {FoodImport.factor=Q4} 0.08048719  0.4101162 1.962611
## 7  {PermanentCropland.factor=Q3,
##     Population.factor=Q1}        => {FoodImport.factor=Q4} 0.07667025  0.7670379 3.670661
## 8  {Population.factor=Q2}        => {FoodImport.factor=Q4} 0.06810862  0.2648969 1.267665
```

```
## 9  {LandArea.factor=Q1,
##     PermanentCropland.factor=Q3,
##     Population.factor=Q1}        => {FoodImport.factor=Q4} 0.06677004  0.7414275 3.548102
## 10 {LandArea.factor=Q1,
##     AgriculturalLand.factor=Q4}  => {FoodImport.factor=Q4} 0.06085908  0.7326101 3.505907
## 11 {ArableLand.factor=Q4}        => {FoodImport.factor=Q4} 0.05876506  0.3003048 1.437109
## 12 {FoodProdIndex.factor=Q1}     => {FoodImport.factor=Q4} 0.05766504  0.2731153 1.306994
## 13 {PopulationPerSqkm.factor=Q1} => {FoodImport.factor=Q4} 0.05763853  0.2330654 1.115335
## 14 {LandArea.factor=Q1,
##     PopulationPerSqkm.factor=Q1} => {FoodImport.factor=Q4} 0.05763853  0.9302674 4.451796
## 15 {ArableLand.factor=Q2}        => {FoodImport.factor=Q4} 0.05599512  0.2874736 1.375705
## 16 {PopulationPerSqkm.factor=Q2} => {FoodImport.factor=Q4} 0.05517342  0.2234208 1.069181
## 17 {LandArea.factor=Q1,
##     ArableLand.factor=Q4}        => {FoodImport.factor=Q4} 0.05451076  0.7160515 3.426666
## 18 {PermanentCropland.factor=Q3,
##     ArableLand.factor=Q4}        => {FoodImport.factor=Q4} 0.05436497  0.6133373 2.935127
## 19 {PopulationPerSqkm.factor=Q3} => {FoodImport.factor=Q4} 0.05342399  0.2129312 1.018982
## 20 {LandArea.factor=Q1,
##     PermanentCropland.factor=Q3,
##     ArableLand.factor=Q4}        => {FoodImport.factor=Q4} 0.05240348  0.8004049 3.830339
## 21 {ArableLand.factor=Q4,
##     AgriculturalLand.factor=Q4}  => {FoodImport.factor=Q4} 0.05048176  0.4374641 2.093485
```

Lets take a look at one of the rules.

```
##   lhs                        rhs                       support confidence     lift
## 1 {LandArea.factor=Q1,
##    Population.factor=Q1} => {FoodImport.factor=Q4} 0.1203796    0.57883 2.769992
```

Interpretation: 'If a country's land area & total population are in the bottom 25% among all the countries, then food imported by that country is going to be in the highest quantile.' It means, a country that has less land and less population, assuming less population who produces food, will have higher food imports. The support is 0.12 which suggests that 12% of countries have less land area & less population and have high food imports. The lift of 2.76 says that if a country has less land area and low population, it is 2.76 times more likely to import more food.

To investigate more, lets filter down our dataframe to contain only such records which satisfy the above rule. i.e. `LandArea.factor == 'Q1'`, `Population.factor == 'Q1'`, `FoodImport.factor == 'Q4'`. After filtering, lets select the columns we want to further investigate. From this set, we chose *Lebanon* to investigate more into its population and food production index over the years.

```
df.10.yrs %>%
  filter(LandArea.factor == 'Q1',
         Population.factor == 'Q1',
         FoodImport.factor == 'Q4') %>%
  select(country_name, funded_year, LandArea.SQKM, LandArea.factor,
         AgriculturalLand.SQKM, AgriculturalLand.factor,
         PermanentCropland.SQKM,  PermanentCropland.factor,
         FoodProdIndex, FoodProdIndex.factor,
         FoodImport.factor, FoodImports,
         TotalPopulation, Population.factor,
         PopulationPerSqkm, PopulationPerSqkm.factor) %>%
  na.omit() %>%
```

```
  unique() %>%
  filter(country_name == 'Lebanon') -> df.rule1.Lebanon
```

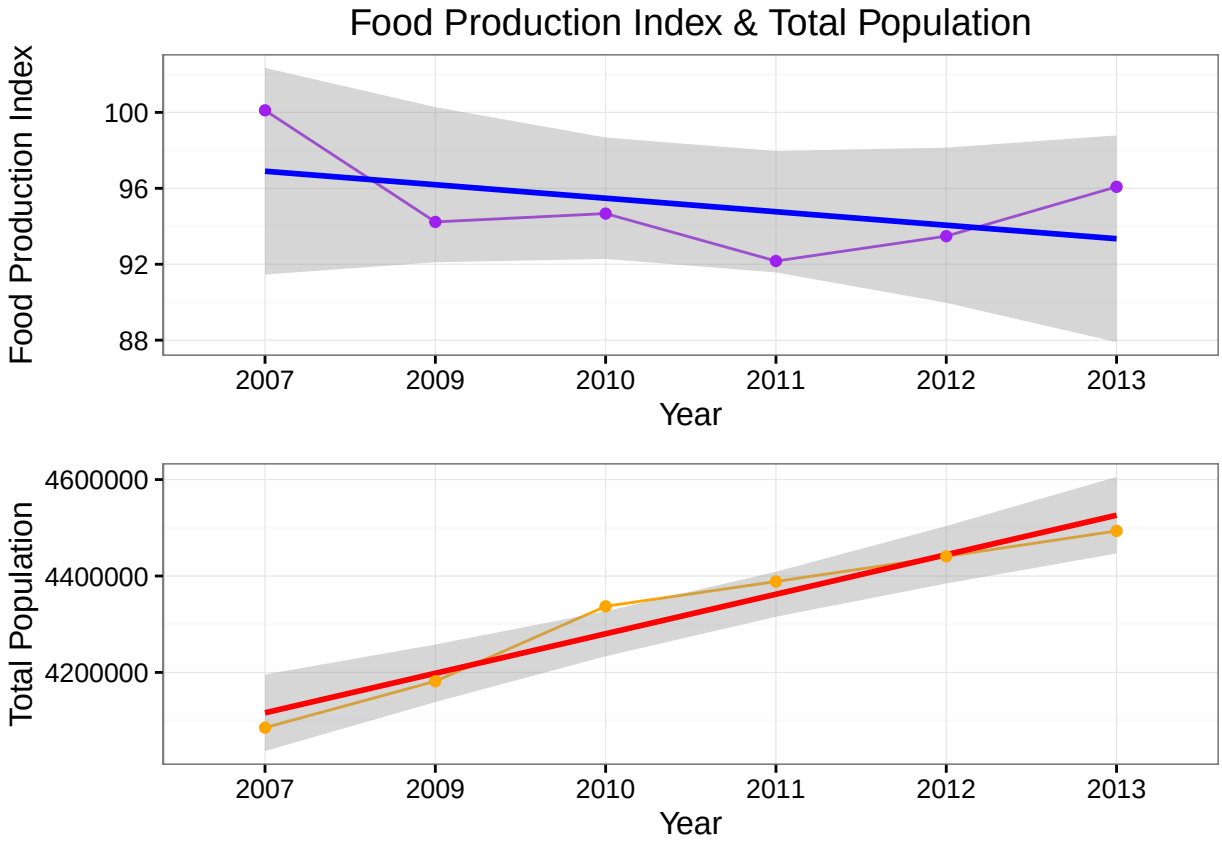
We plot 2 line graphs for food production index & total population of Lebanon from 2007 to 2013. This helps us understand the trend that has been going on in this country.

```
p1 <- ggplot(df.rule1.Lebanon, aes(funded_year, FoodProdIndex, group = 1)) +
  geom_line(color="purple") +
  stat_smooth(method="lm", color="blue") +
  geom_point(color="purple") +
  theme_bw() +
  labs(title = "Food Production Index & Total Population",
       x="Year",
       y="Food Production Index")

p2 <- ggplot(df.rule1.Lebanon, aes(funded_year, TotalPopulation, group = 1)) +
  geom_line(color="orange") +
  stat_smooth(method="lm", color="red") +
  geom_point(color="orange") +
  theme_bw() +
  labs(x="Year",
       y="Total Population")
```

We plot the 2 line graphs together with the function `rbind` and `grid.draw`.

```
grid.newpage()
grid.draw(rbind(ggplotGrob(p1), ggplotGrob(p2), size = "last"))
```



Interpretation: We see that there is a steady growth in the country's population and an overall decrease in the food production index. If we assume Lebanon to represent the group of countries which have land area and population in lower quantile, we can say that a steady increase in population and a decrease in food production index is the reason why their food imports are in the highest quantile.

Lets take a look at another rule.

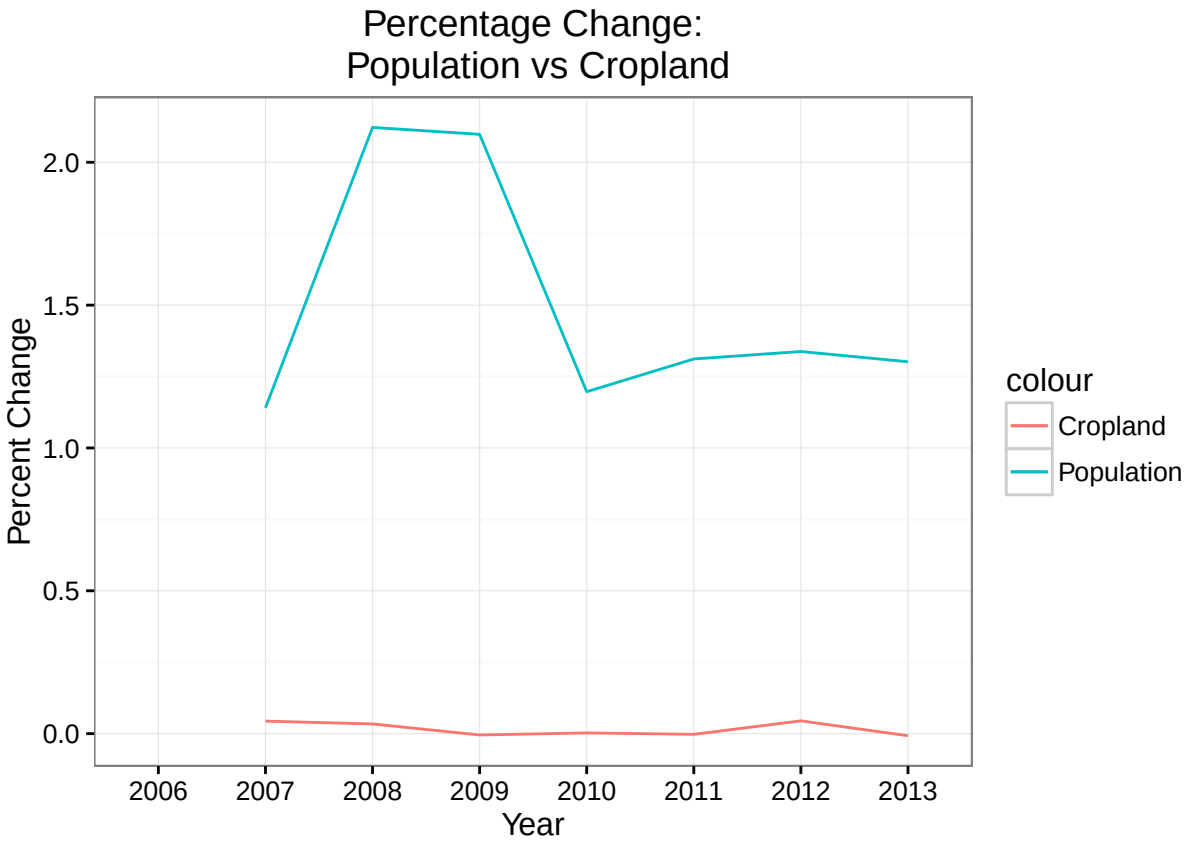
```
  inspect(sort(df.rules.fi, by = "support")[20])
```

```
##   lhs                                rhs                       support confidence     lift
## 1 {LandArea.factor=Q1,
##    PermanentCropland.factor=Q3,
##    ArableLand.factor=Q4}        => {FoodImport.factor=Q4} 0.05240348  0.8004049 3.830339
```

Interpretation: 'If a country's land area is in the bottom quantile, but its permanent cropland and arable land are in the top half, then food imported by that country is going to be in the highest quantile.' The support is 0.05 which suggests that 5% of countries that have less land area than other countries have highest food imports. The lift of 3.83 says that if a country has less land area, it is 3.83 times more likely to import more food.

One such country with `LandArea` in Q1, `PermanentCropland` in Q3 and `ArableLand` in Q4 is *Azerbaijan*. We assume that Azerbaijan can represent the above rule. From the dataframe `df.10.yrs`, we filter `LandArea.factor` = Q1, `PermanentCropland.factor` = Q3 and `ArableLand.factor` = Q4. Then select the relevant columns needed, omit null values (NAs), take uniques and filter for country Azerbaijan. Then we calculate the percentage change in permanent cropland and population for the country over the years.

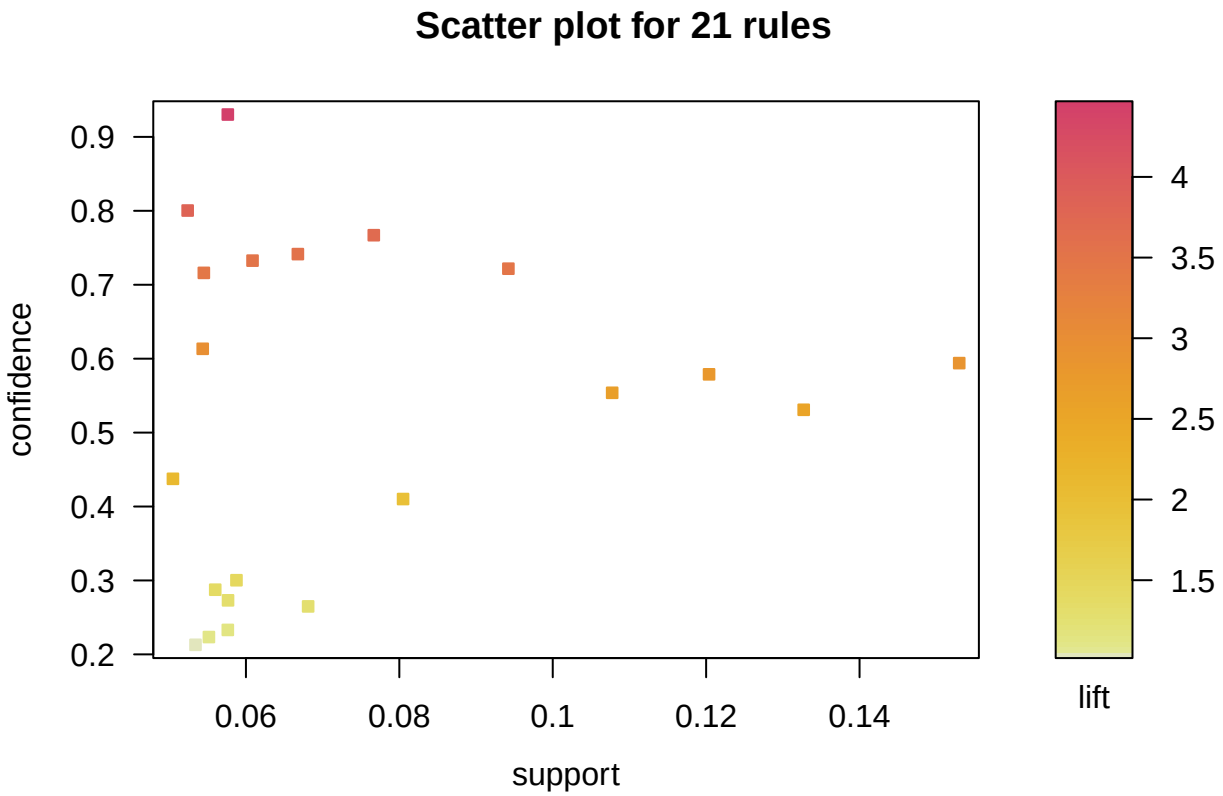
```
df.10.yrs %>%
  filter(LandArea.factor == 'Q1',
         PermanentCropland.factor == 'Q3',
         ArableLand.factor == 'Q4') %>%
  select(country_name, funded_year, LandArea.SQKM, LandArea.factor,
         AgriculturalLand.SQKM, AgriculturalLand.factor,
         PermanentCropland.SQKM,  PermanentCropland.factor, `PermanentCropland%`,
         FoodProdIndex, FoodProdIndex.factor,
         FoodImport.factor, FoodImports,
         TotalPopulation, Population.factor,
         PopulationPerSqkm, PopulationPerSqkm.factor,
         ArableLand.factor, ArableLand.SQKM, `ArableLand%`) %>%
  na.omit() %>%
  unique() %>%
  filter(country_name == 'Azerbaijan') %>%
  select(funded_year, TotalPopulation, `PermanentCropland%`) %>%
  mutate(Percentage_Change_pop = ((TotalPopulation-lag(TotalPopulation))/lag(TotalPopulation))*100) %>%
  mutate(Percentage_Change_crop = `PermanentCropland%`-lag(`PermanentCropland%`)) %>%
  select(funded_year, Percentage_Change_pop, Percentage_Change_crop) %>%
  ggplot(aes(funded_year, group = 1, color="red")) +
  geom_line(aes(y = Percentage_Change_crop, color='Cropland')) +
  geom_line(aes(y = Percentage_Change_pop, color='Population')) +
  theme_bw() +
  labs(title = "Percentage Change:\n Population vs Cropland",
       x = "Year",
       y="Percent Change")
```



Interpretation: From year 2007 to 2008, there was a 2% population increase in Azerbaijan but at the same time the permanent cropland saw a decline. Later on, the population saw a decline and then again a constant incline, but permanent cropland kept decreasing. From this we can infer that, since permanent cropland of Azerbaijan is decreasing steadily, the country has to import more foods and thus has a higher percentage of food imports out of its total merchandise imports.

Then we plot all these rules to see their distribution

```
plot(df.rules.fi)
```



Interpretation: There are very few rules which have good support. But almost half the rules show good lift and confidence. This just means that there are many different scenarios leading to high food imports. Not all of them have high support, but given the variability of the data, this is not unexpected.

Conclusion

We are examining the Kiva data set and WDI data set to find the money being spent on food by the people across the world. We are analysing the merged data on the food import expenditure of a country and are trying to find the reason for the high and low numbers for different countries. We are examining the data on Food Production Index and Amount spent on Agricultural development in the country. Through this we are trying to establish the reasons for the high amount being spent on food imports. In our analysis, we are taking top and bottom 5 countries based on the %-FoodImport for year 2014. We made 2 different dataframes for top and bottom countries.

The following are our major conclusions:

• India and United States have lower percent of food imports while Benin has the highest percentage of food imports.

• Benin imports more food even though it has more food production index. This is mainly because of the weather conditions in Benin. Crops do not survive due to the weather and hence the import is more.

• A country that has less agricultural land will have higher food imports

• If a country's land area and population are in the bottom 25% among all the countries, then food imports for that country is going to be in the highest quantile.

• Azerbaijan, a country with land area in the lowest quartile has permanent cropland in the top half. But still the country's food imports are high because of the steady increase in the population.