INFO7374 - Kiva & World Development Indicators

Bowei Wang, Dongyue Li, Sarthak Agarwal, Sriram Chandramouli

7 June 2016

Contents

1. Introduction **2**

Analysis Goals . . . 2

2. Data Profile **2**

2.1 Dataset description . . . 2

2.1 Description of rows and observations . . . 3

3. Dataset preparation **4**

3.1 Load libraries . . . 4

3.2 Loading the Kiva loans dataset into dataframe . . . 4

3.3 Loading the WDI dataset into dataframe . . . 5

3.4 Create new variables . . . 6

`cleanedyear` . . . 6

`countryF` . . . 6

`loan_amount_numeric` . . . 6

`yearF` . . . 6

`funded_date_cleaned` . . . 6

`paid_date_cleaned` . . . 7

`sector_F` . . . 7

3.5 Merging Kiva and WDI dataframes . . . 7

4. Variable summaries and visualizations **7**

4.1 Total loan amount by country . . . 7

4.2 Loan requirement by country . . . 8

4.3 Loan amount and sector . . . 10

4.4 `borrowers.genderF` . . . 11

4.5 Loan amount, gender and sector . . . 12

4.6 Country's GDP and their loan amount requirement . . . 13

4.7 Average loan amount, paid amount and sector . . . 19

4.8 Loan amount, Status and Sector . . . 20

4.9 The time from funded date to payback date by top 10 loan amount countries . . . 22

5. Relationships between variables **23**

Relationship between status and other variables 31

6. Conclusion **34**

1. Introduction

A microfinance institution is an organization that is a source of financial services for small businesses lacking access to traditional banking systems. Kiva (www.kiva.org) is a microlending website that works with such microfinance institutions to raise funds for low income entrepreneurs and provide loans for their business. Through this platform, anyone with an internet connection can make an interest free loan.

Kiva has field partners which are mainly microfinance institutions but also schools, NGOs, social enterprises. These field partners work at the local level and disburse loans to the borrowers. Loans are pre-disbursed which means that the loan is given out before being funded on the Kiva website. The field partners then collect stories, photos, and videos from the borrower and post them on Kiva which are published on the website after review. Anyone who can access the Kiva website can browse through the borrower's profile to make a loan for at least 25$. Kiva aggregates all the money and backfills the loan already disbursed by the field partner. Field partners then collect repayments from borrowers. Some microfinance institutions may charge an interest rate but around 60% of the partners are non-profits. The amount is then repaid to the Kiva lender.

Every year World Bank releases a collection of World Development Indicators (WDI). The data represents the economic, demographic, social, environmental, educational and cultural indicators. It contains over 800 indicators covering more than 150 countries representing the most current and accurate global development data available which includes national, regional and global estimates.

Analysis Goals

Kiva complete business model is based around lending and borrowing loans. Kiva's main focus is to be a medium between the loan borrowers and lenders. Hence lenders are very important for Kiva and eventually to loan borrowers. Our objective of this analysis is to help Kiva and loan lenders to make better decisions that would help in the economic development of the countries and help alleviate poverty. In this report we analyze the loan amount, and its relationship with other variables such as sectors, genders and countries. We also analyze the loan amount funded and the repayment trends of these loans.

We further analyze the loans funded across countries with the GDP of those countries and provide information about the countries which needs more attention than others. [Make better sentences]

2. Data Profile

In the following two sub sections we understand the data and it's structure. We further identify the relevant variables within the Kiva dataset that are important for our analysis.

2.1 Dataset description

The Kiva dataset consists of information on lenders and loans. We work primarily with the loans dataset and reference the lenders dataset where necessary. The data is available at www.build.kiva.org website and has three sub directories - `lenders`, `loans` and `loans_lenders`.

To analyze the structure, we load the 2129 files from the `loans` sub directory.

2.1 Description of rows and observations

Each loan entry in Kiva has the following 30 variables:

```
##  [1] "id"                                "name"
##  [3] "description"                       "status"
##  [5] "funded_amount"                     "basket_amount"
##  [7] "paid_amount"                       "image"
##  [9] "video"                             "activity"
## [11] "sector"                            "themes"
## [13] "use"                               "delinquent"
## [15] "location"                          "partner_id"
## [17] "posted_date"                       "planned_expiration_date"
## [19] "loan_amount"                       "lender_count"
## [21] "currency_exchange_loss_amount" "bonus_credit_eligibility"
## [23] "tags"                              "borrowers"
## [25] "terms"                             "payments"
## [27] "funded_date"                       "paid_date"
## [29] "journal_totals"                    "translator"
```

For our analysis, we choose the following variables and create few additional variables:

Variable	Type	Description
`location.country`	Character	The country of the loan borrower
`CountryF`	Character	Country variable converted to factor variable
`borrowers.gender`	Character	Gender of the borrower
`borrowers.GenderF`	Character	Gender variable converted to factor variable
`loan_amount`	Number	The amount of loan taken by the borrower
`loan_amount_numeric`	Number	Country variable converted to factor variable
`status`	Character	Loan payment status
`statusF`	Character	Status variable converted to factor variable
`sector`	Character	Economic sector of the loan entry
`Sector_F`	Character	Sector variable converted to factor variable
`paid_amount`	Number	The amount of money paid back by the borrower
`posted_date`	Character	Date on which the loan is posted on Kiva
`cleanedyear`	Numeric	Year extracted from the posted_date
`YearF`	Character	cleanedyear variable converted to factor variable
`funded_date`	Character	Date on which loan is funded
`funded_date_cleaned`	Date	funded_date converted to YYYY-MM-DD format
`paid_date`	Character	Date on which loan is paid off
`paid_date_cleaned`	Date	paid_date converted to YYYY-MM-DD format
`day_diff`	Numeric	Difference between funded date and paid date
`activity`	Character	Reason for which loan is taken
`lender_count` [to be removed]	Number	Number of people who lended money

Units of these variables are:

Variable	Unit
`loan_amount`	Dollar
`loan_amount_numeric`	Dollar
`paid_amount`	Dollar

Variable	Unit
`paid_amount_numeric`	Dollar

3. Dataset preparation

To prepare the data for analysis, we:

1. load the necessary libraries,
2. load the Kiva loans dataset into dataframe,
3. load the WDI dataset into dataframe,
4. create new variables,
5. merge Kiva and WDI dataframe

3.1 Load libraries

Following libraries are used in this report:

- dplyr
- magrittr
- RJSONIO
- rlist
- WDI
- ggplot2
- parallel
- ffbase
- reshape2
- gridExtra
- gapminder
- grid
- maps
- arules

3.2 Loading the Kiva loans dataset into dataframe

In the next few steps we load the JSON files under the Kiva `loans` sub directory into R. Since there are more than 1 millions loan entries, we use parallel computing to load the complete data.

We create a function which generates a dataframe from a list.

```
dfrow.from.list = function(aList) {
  data.frame(rbind(unlist(aList)),
             stringsAsFactors=FALSE)
}
```

We use the above function and create another function to read a JSON file into a dataframe. We use `isValidJSON` function to skip invalid JSON files.

```
readJSONFileIntoDataFrame <-
  function (filename) {
    if(isValidJSON(paste(data.folder,
          filename,
          sep=""))){
    paste(data.folder,
          filename,
          sep="") %>%
      fromJSON() %>%
      { .$loans } %>%
      list.select(posted_date, location.country = location$country, sector, loan_amount,
                  funded_date, paid_date, status, lender_count, activity,
                  borrowers = borrowers[1], paid_amount ) %>%
      lapply(dfrow.from.list) %>%
      bind_rows()
    }
  }
```

We create a socket cluster which creates a set of copies of R running in parallel. Based on the system configuration we use four cores and use the function `clusterExport` which provides several ways to parallelize computations.

```
cl <- makeCluster(4)
clusterExport(cl,
              c('dfrow.from.list','isValidJSON', 'data.folder', '%>%',
                'list.select','bind_rows','fromJSON',
                'readJSONFileIntoDataFrame'))
```

We use parallel version of Lapply to apply `readJSONFileIntoDataFrame` function on JSON files. Then stop cluster when finished.

```
create.loan.df.cl = function(cl, loan.file.in) {
  loan.file.in %>%
    { parLapply(cl, ., readJSONFileIntoDataFrame) } %>%
    bind_rows()
}
loan.df = create.loan.df.cl(cl, loan.file[1:40])
stopCluster(cl)
```

3.3 Loading the WDI dataset into dataframe

We choose the following Indicator code from the WDI dataset. We add it to the `indicator.codes` variable.

```
indicator.codes = c("NY.GDP.MKTP.CD")
```

We read the WDI data for this code into the `df` dataframe.

```
df <- WDI(indicator = indicator.codes, extra = TRUE)
```

3.4 Create new variables

Following new variables are created for our analysis.

- cleanedyear
- countryF
- loan_amount_numeric
- yearF
- funded_date_cleaned
- paid_date_cleaned
- sector_F

cleanedyear

We create a numeric variable called `cleanedyear` by extracting year from `posted_date`. This variable is used in merging the Kiva and WDI dataset.

```
cleanedyear <- substr(loan.df[['posted_date']],1,4)
loan.df[["cleanedyear"]] <- as.numeric(cleanedyear)
summary(loan.df[["cleanedyear"]])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    2006    2012    2012    2012    2012    2012     135
```

We observe that the loan data is available from 2006 to 2013 and most of the loans are in the later years.

countryF

We create this variable by converting `location.country` into a factor variable.

```
loan.df$countryF <- factor(loan.df$location.country)
```

loan_amount_numeric

We create this variable by converting the `loan_amount` variable, which is character to numeric.

```
loan.df$loan_amount_numeric <- as.numeric(loan.df$loan_amount)
```

yearF

We create this variable by converting the `cleanedyear` variable into a factor variable.

```
loan.df$yearF <- factor(loan.df$cleanedyear)
```

funded_date_cleaned

We create this variable from the `funded_date` variable by removing unnecessary time information and extracting the date in the format YYYY-MM-DD.

```
loan.df$funded_date_cleaned <-
  as.Date(substr(loan.df$funded_date, 1, 10), "%Y-%m-%d")
```

paid_date_cleaned

We create this variable from the `paid_date` variable by removing the unnecessary time information and extracting the date in the format YYYY-MM-DD.

```
loan.df$paid_date_cleaned <-
  as.Date(substr(loan.df$paid_date, 1, 10), "%Y-%m-%d")
```

sector_F

We create this variable by converting `sector` variable into a factor variable.

```
loan.df$sector_F <- factor(loan.df$sector)
```

3.5 Merging Kiva and WDI dataframes

We use the `cleanedyear` and `location.country` variables in loan dataframe and join them on the basis of `country` and `year` variables in WDI dataframe. We use left join to gather all the data from Kiva dataset after merging.

```
finaldf <-
  merge(loan.df, df, by.x = c("location.country", "cleanedyear"),
        by.y = c("country", "year"), all.x = TRUE
  )
```

4. Variable summaries and visualizations

In the next few sections, we analyze the significant single and multiple variables for our analysis.

4.1 Total loan amount by country

Since our analysis is primarily focused on loan amount and its relationship with other variables, loan amount plays a significant role in our analysis. We visualize the loans and its distribution across various countries by plotting the loan amount on a world map.

We load the map data from `world2` and exclude Antarctica from the map.

```
world <- map_data("world2")
world <- subset(world,region!="Antarctica")
```

We calculate the total loan amount for each country.

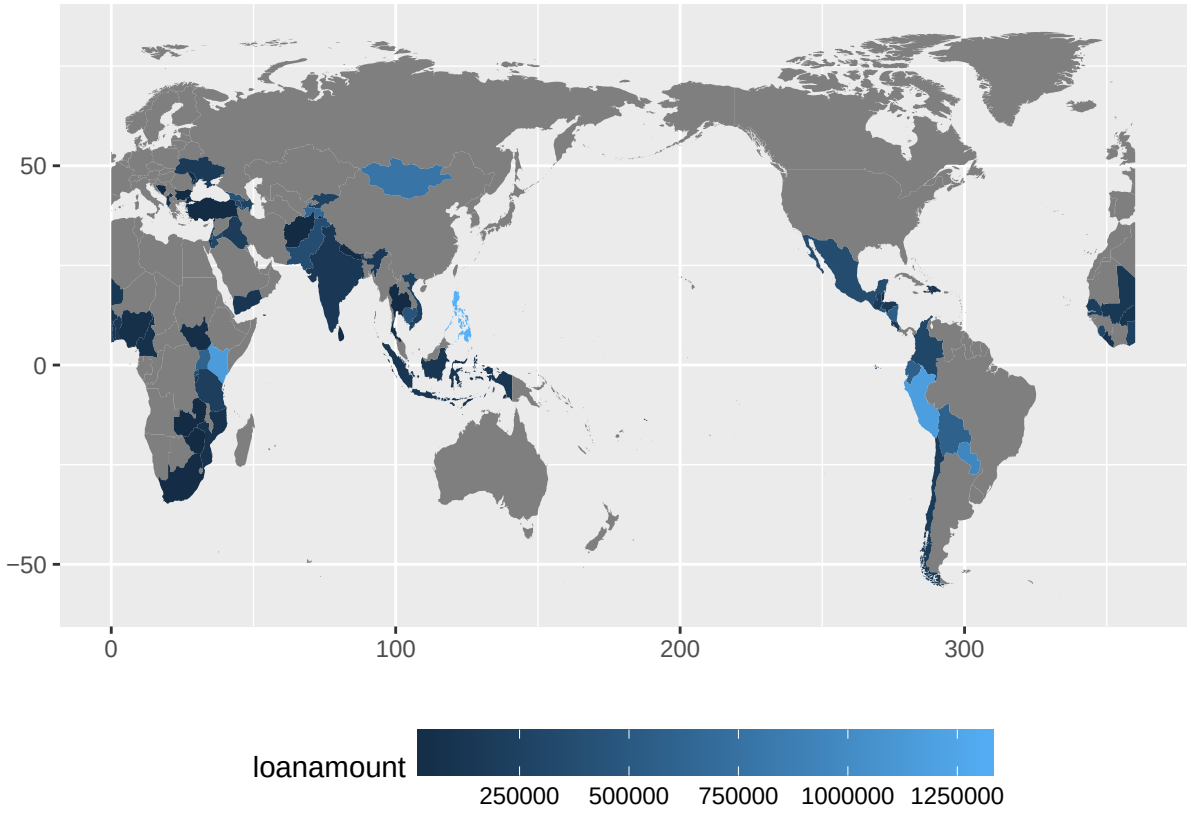
```
loan_amount_by_country <- condSum(loan.df$loan_amount_numeric,
                                  loan.df$countryF,
                                  na.rm = FALSE)
```

We match the country name with `loan_amount_by_country` variable.

```
world$loanamount <- loan_amount_by_country[
  match(world$region,
        names(loan_amount_by_country),
        nomatch=NA)]
```

We fill the data with the total loan amount by country and move legend position to bottom of the graph.

```
map <- qplot(long, lat, data = world,
             group = group, fill = loanamount,
             geom ="polygon",ylab="",xlab="")
map + theme(legend.position="bottom",
            legend.key.width = unit(3, "line"))
```



We observe that amongst 250 countries for which loans are posted on Kiva website, maximum loan requirement is in developing countries of South America, Africa and Asia. Our focus is on these countries and providing insights that will help in redirecting more money to these countries.

4.2 Loan requirement by country

We calculate the top 10 countries that have the most loan requirement for the year range 2006-2011. This will help lenders to focus on lending money to these specific countries which have the maximum need.

We calculate total loan amount for every country and year.

```
loan_amount_by_country <-
  condSum(loan.df$loan_amount_numeric,
          list(loan.df$countryF,loan.df$yearF),
          na.rm = FALSE)
```

We create a dataframe of the above matrix.

```
loan_amount_by_country_df <-
  data.frame(row.names(loan_amount_by_country),
             loan_amount_by_country)
```

We subset the above dataframe for the year range 2006-2011 and rename columns for readability.

```
loan_amount_by_country_df_from2006_to2011 <-
  loan_amount_by_country_df[,c("row.names.loan_amount_by_country.",
                               "X2006","X2007","X2008","X2009",
                               "X2010","X2011")]
colnames(loan_amount_by_country_df_from2006_to2011) <-
  c("country", "2006", "2007","2008",
    "2009", "2010","2011")
```

We calculate total loan amount for each country from 2006-2011.

```
loan_amount_by_country_df_from2006_to2011$total <-
  rowSums(loan_amount_by_country_df_from2006_to2011[2:7])
```

We sort this dataframe on the basis of total loan amount and get the records for top 10 countries which have the most total loan amount.

```
loan_amount_by_country_df_from2006_to2011_top10 <- head(arrange(
  loan_amount_by_country_df_from2006_to2011,
               desc(total)), n = 10)
loan_amount_by_country_df_from2006_to2011_top10
```

```
##        country 2006  2007  2008  2009  2010  2011  total
## 1         Peru    0  9775 34200 44400 52650 45425 186450
## 2     Cambodia    0 10600 31450 29500 34225 37400 143175
## 3  Philippines    0     0   325 26325 48025 57075 131750
## 4      Bolivia    0  3700 17625 33025 22025 37675 114050
## 5       Uganda    0     0 30450 29350 26675 25975 112450
## 6        Kenya 4400  9550  1800  4550 24600 50025  94925
## 7    Nicaragua    0  1375 10625 23175 27725 20400  83300
## 8   Tajikistan    0  5350 11800 20775 11450 27875  77250
## 9      Ecuador 1600  2325  4350   675 23425 37650  70025
## 10  Azerbaijan  800 13325 16275 14625 13250  6350  64625
```

We see that Peru, Cambodia and Phillipines are the countries that have the most loan requirement. Hence lenders can filter the loan postings on Kiva website by these countries.

Also, it is important to note that for Cambodia, Bolivia and Nicaragua loan requirement increases till 2010 and then decreases in 2011. This can be due to economic stagnation or policy change in that country.

4.3 Loan amount and sector

This analysis provides an understanding of the total loan amount for each sector. There are 15 sectors for which loans are posted on Kiva website.

We calculate the total loan amount for each sector.

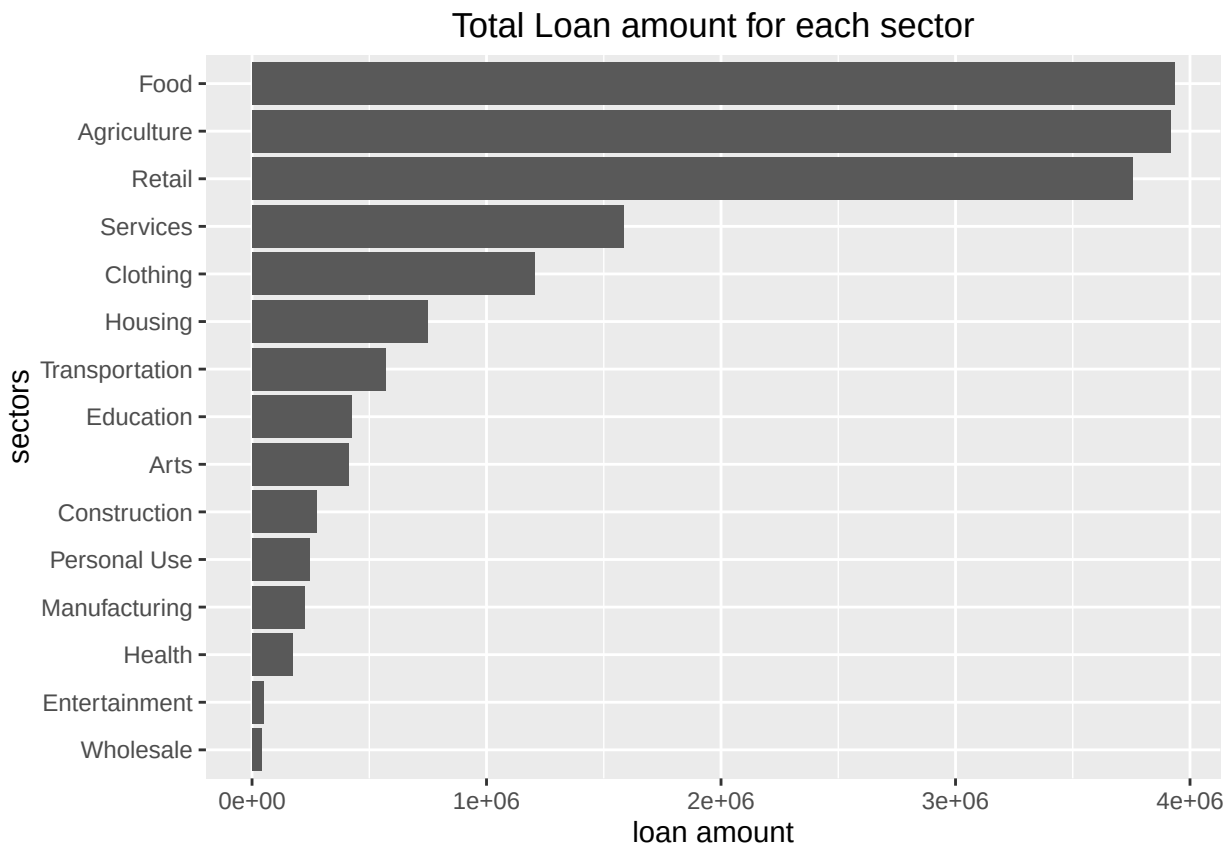
```
loan_amount_by_sector <- condSum(loan.df$loan_amount_numeric,
                                 loan.df$sector_F,na.rm = FALSE)
```

We create the `loan_amount_by_sector_df` dataframe from the `loan_amount_by_sector` variable created above.

```
loan_amount_by_sector_df <-data.frame(names(loan_amount_by_sector),
                                      loan_amount_by_sector)
```

The code chunk below plots the graph for total loan amount for each sector and sort it on the basis of loan amount.

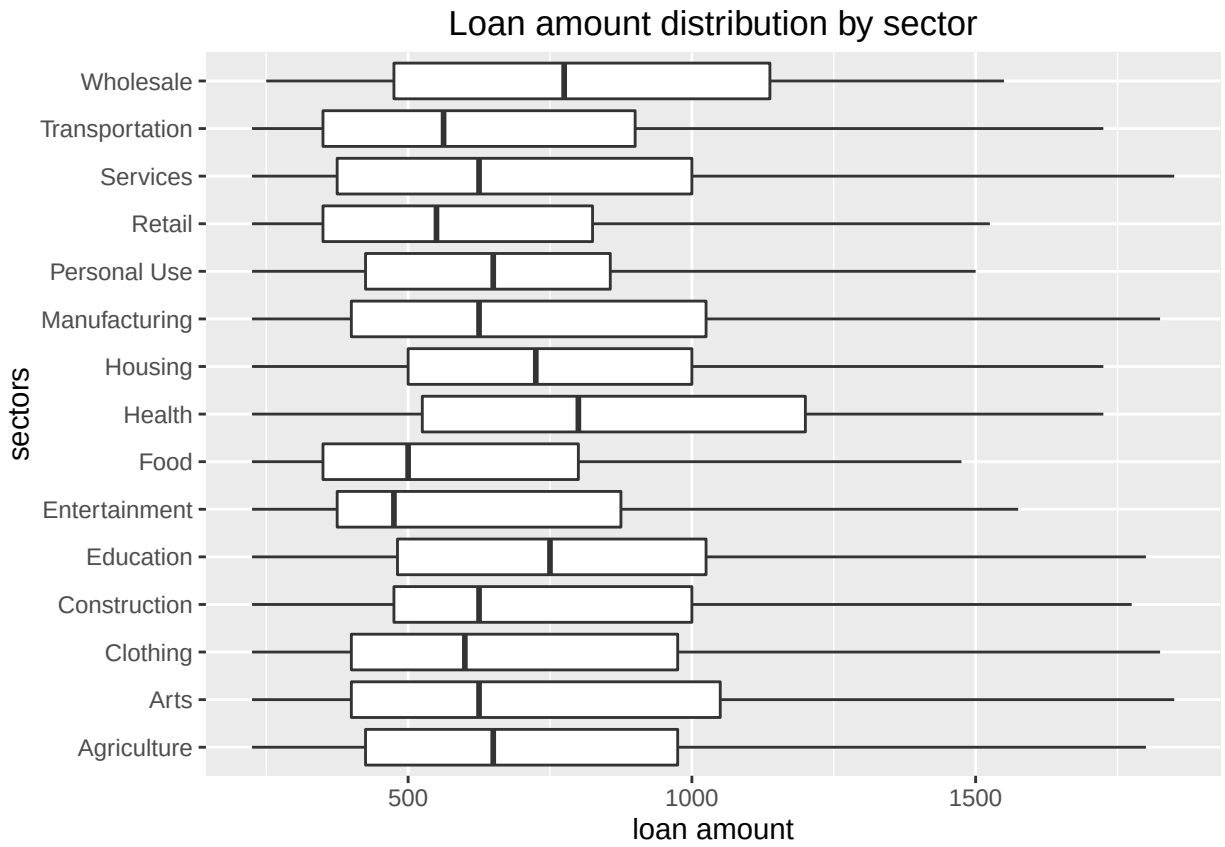
```
loan_amount_by_sector_df %>%
  ggplot (aes(x = reorder(names(loan_amount_by_sector),
                          loan_amount_by_sector), y = loan_amount_by_sector)) +
  ggtitle("Total Loan amount for each sector") +
  xlab("sectors") +
  ylab("loan amount") +
  geom_bar(stat = "identity") +
  coord_flip()
```



We observe that people in the agriculture, food and retail sectors are asking for the most amount of loans. Lenders should focus on lending money to the loans in these sectors. Entertainment and wholesale sectors have the least loan requirement.

To further analyze the total loan amount distribution by sector we use a box plot. We modify the wide scale to 10 ~ 90 percentile to eliminate extreme variables.

```
loan.df %>%
ggplot(aes(sector_F,loan_amount_numeric)) +
ggtitle("Loan amount distribution by sector") +
xlab("sectors") +
ylab("loan amount") +
geom_boxplot(outlier.shape = NA) +
scale_y_continuous(limits = quantile(loan.df$loan_amount_numeric, c(0.1, 0.9))) +
coord_flip()
```



We could see that wholesale has the highest average loan amount and low variation as compared to other sectors. Food and entertainment sectors have high variation.

4.4 `borrowers.genderF`

The `borrowers.genderF` variable represents the loan borrower's gender. We create this variable by converting `borrowers.gender` into a factor variable.

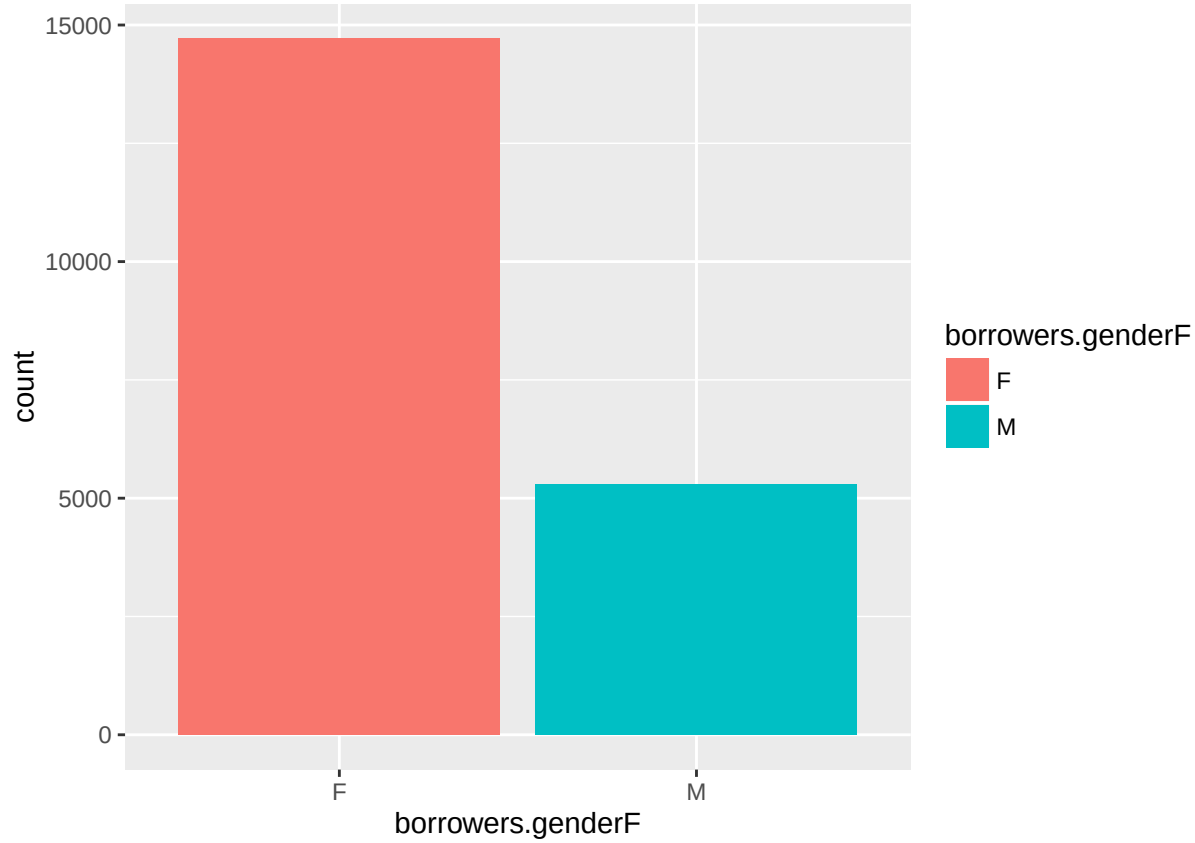
```
loan.df$borrowers.genderF <- factor(loan.df$borrowers.gender)
```

The code below prints the summary statistics and a bar graph for visual representation.

```
summary(loan.df$borrowers.genderF)
```

```
##     F     M
## 14713  5287
```

```
loan.df %>%
  ggplot(aes(x=borrowers.genderF)) +
  geom_bar(aes(fill=borrowers.genderF))
```

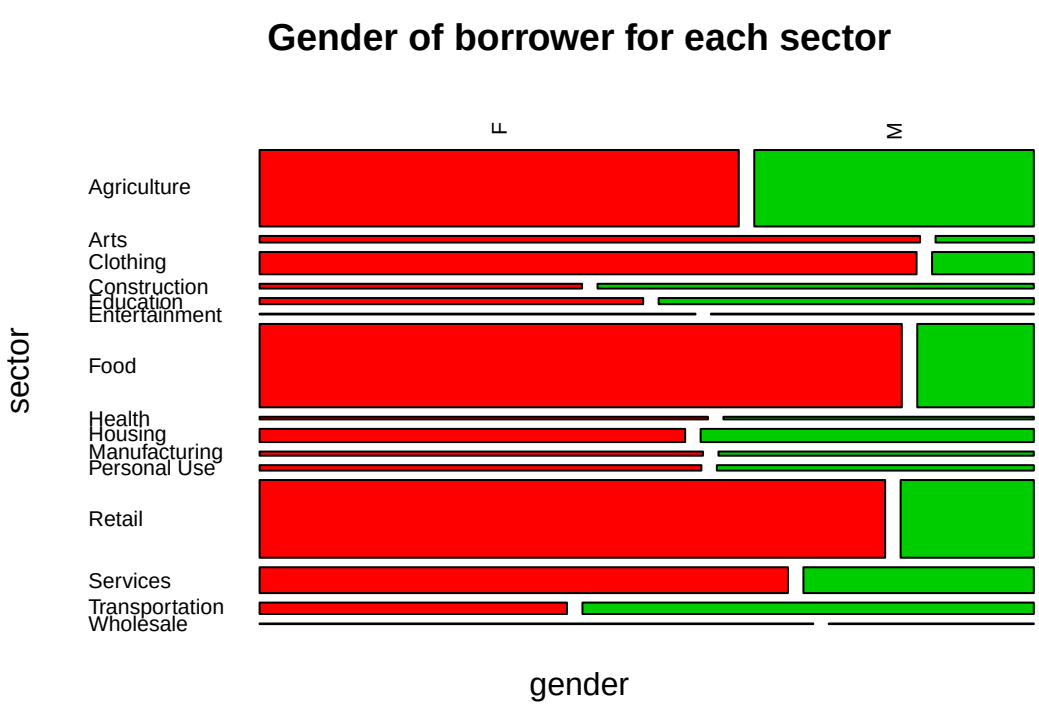


We observe that around 70 percent of the loans posted on the Kiva website are by females.

4.5 Loan amount, gender and sector

We analyze the relationship between the total loan amount, gender and sector. Our goal is to visualize the gender wise distribution of loan amount of every sector. This will enable the lenders to help the society in achieving gender equality. Below code plots a mosaic graph which represent the gender on the x-axis, sector on y-axis and the width of the bars as loan amount.

```
mosaicplot(sector_F ~ borrowers.genderF, data = loan.df,
           main = "Gender of borrower for each sector",
           xlab = "gender", ylab = "sector",
           dir = c('h','v'), las = 2,
           col = c(2:9))
```



We observe that despite 70 percent of the total loans posted are by females, some sectors like Construction, Transportation and Manufacturing have more loan postings by males.

4.6 Country's GDP and their loan amount requirement

We analyze the relationship between the total loan amount for the countries with higher loan requirement and the GDP of those countries. As Kiva aims to alleviate poverty from the developing countries, this analysis helps Kiva and lenders to contribute towards the country's development by making better decision of lending loans.

We conducted this analysis for a year range of 2006 to 2011.

We drop all null `posted_date` records from the loans dataframe. We observe that in some records `posted_date` is null but `loan_amount` is not null. We remove the records with null `posted_date` before adding the `loan_amount` by year.

```
loan.df <- loan.df[!is.na(loan.df$cleanedyear),]
```

We create a dataframe called `GDP_df` with GDP, year and top 10 countries which have the most loan amount requirement.

```
GDP_df <- data.frame(df$year, df$NY.GDP.MKTP.CD, df$country)
```

We merge the above dataframe with the WDI dataframe by year and country. We change the data structure by converting all year columns into rows and converting the data to `long` format using `melt()` package.

```
loan_amount_by_country_df_from2006_to2011_top10_long <-
  melt(loan_amount_by_country_df_from2006_to2011_top10[1:7], id="country")
```

We plot different graphs based on the countries and observe the relationship between GDP and loan amount over the years.

We merge `loan_amount_by_country_df_from2006_to2011_top10_long` dataframe which has the top 10 countries with the most amount of loan requirement for the year 2006-2011 and `GDP_df` dataframe created above by `country` and `year`.

```
loan_GDP_df <- merge(loan_amount_by_country_df_from2006_to2011_top10_long,
                     GDP_df, by.x = c("country", "variable"),
                     by.y = c("df.country", "df.year"))
```

We rename the columns of the `loan_GDP_df` for better readability.

```
colnames(loan_GDP_df) <- c("country", "year", "loanamount","GDP")
```

We convert `loanamount` and `GDP` columns into rows using melt function.

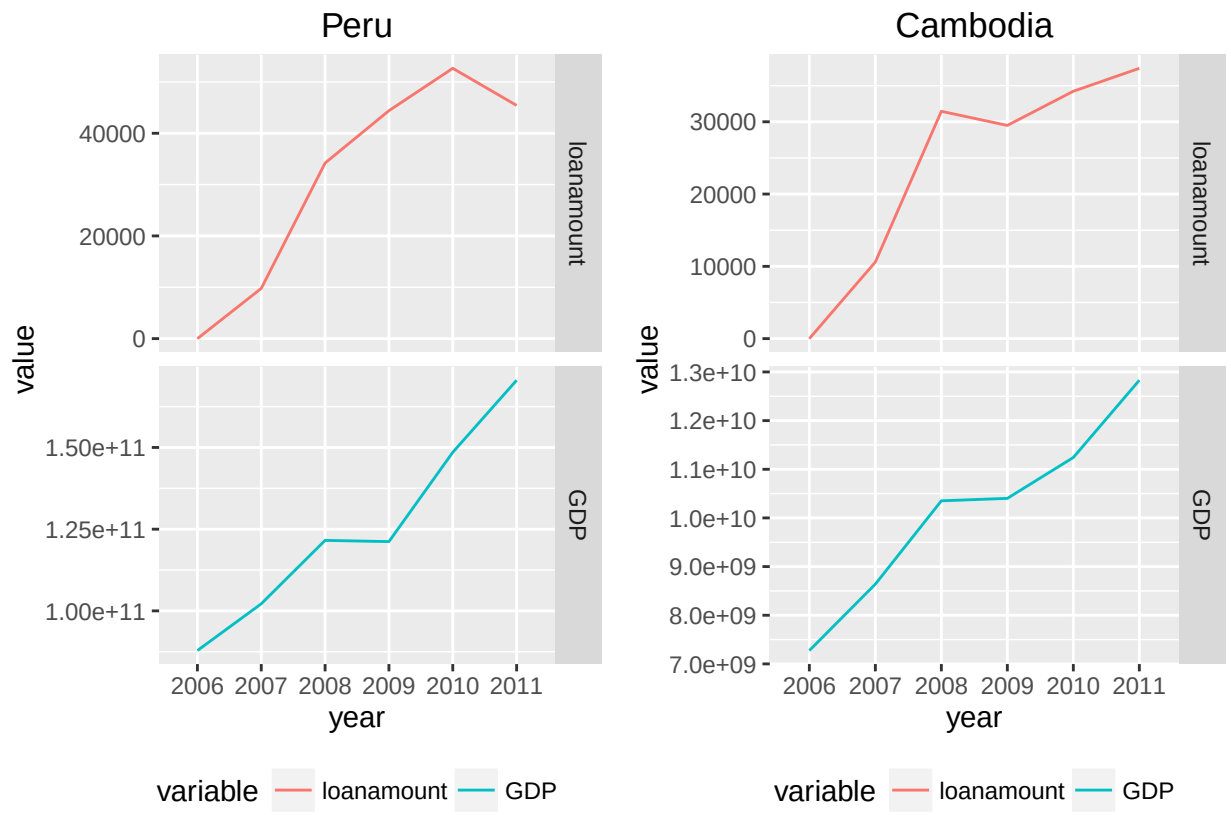
```
loan_GDP_df <- melt(loan_GDP_df[1:4], id=c("country","year"))
```

We get the country names and store it in a variable `top10_country_name`.

```
top10_country_name <-
  loan_amount_by_country_df_from2006_to2011_top10$country %>%
  sapply(as.character)
```

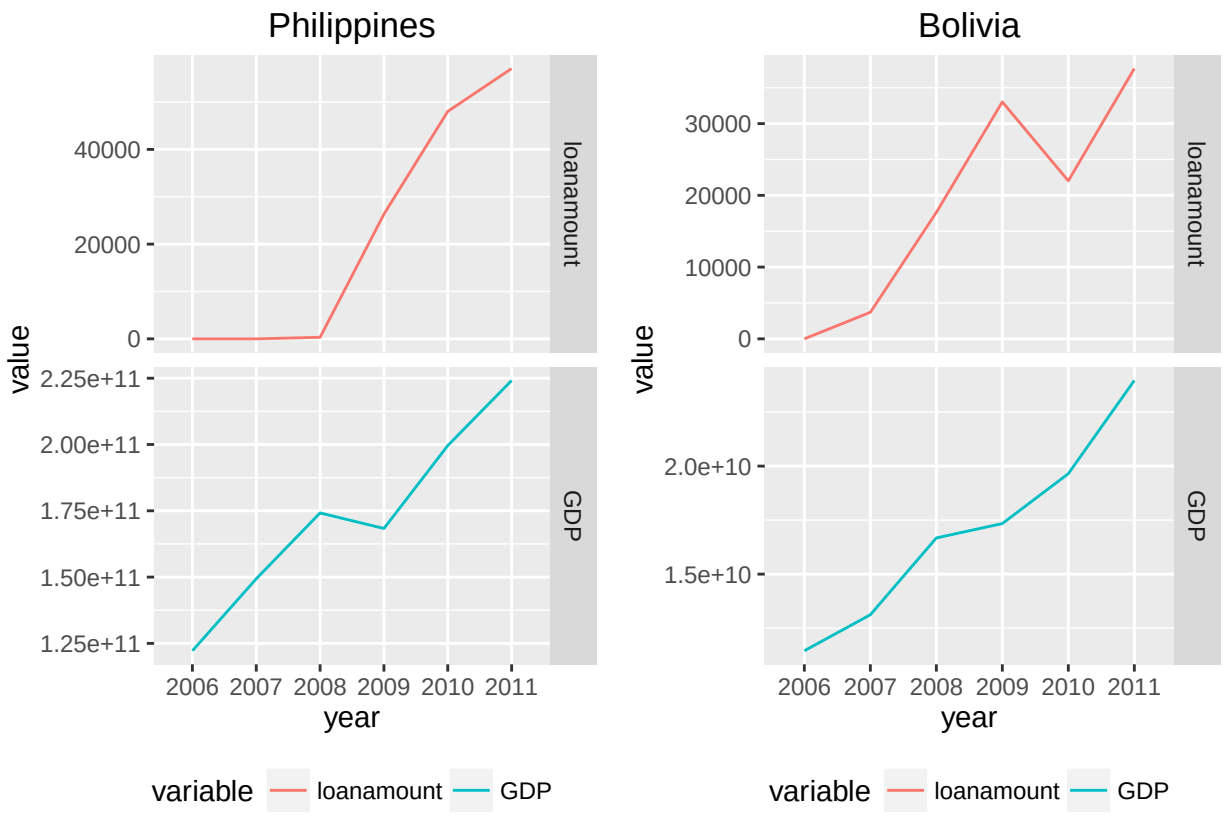
```
title <- top10_country_name[1]
loan_GDP_df_filter <- filter(loan_GDP_df, country %in% top10_country_name[1])
p1 = loan_GDP_df_filter %>%
  ggplot(aes(x=year, y=value, group=variable,
             colour=variable)) +
  geom_line()+
  ggtitle(title) +
  xlab("year") +
  ylab("value")
p1 = p1+theme(legend.position="bottom") + facet_grid(variable ~ ., scales = "free_y")
```

```
grid.arrange(p1, p2, ncol=2)
```



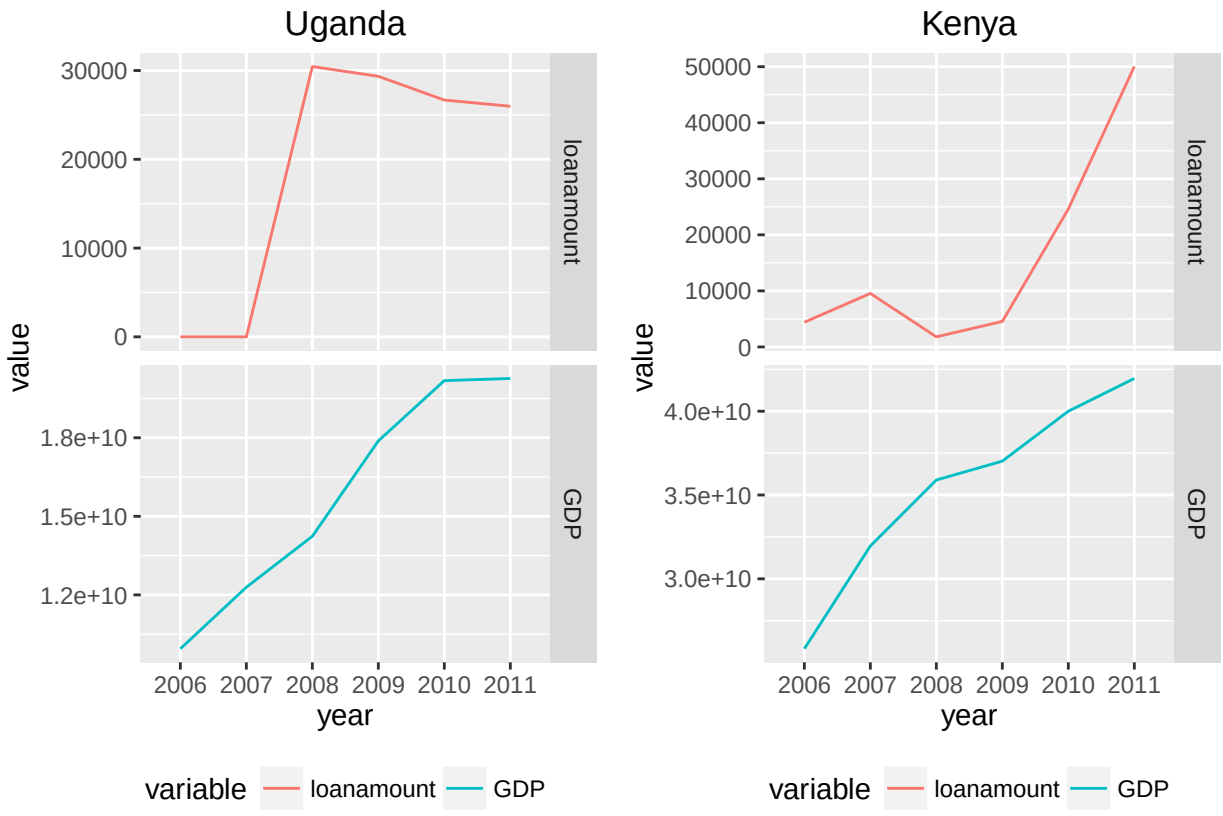
[Write takeaway]

```
grid.arrange(p3, p4, ncol=2)
```



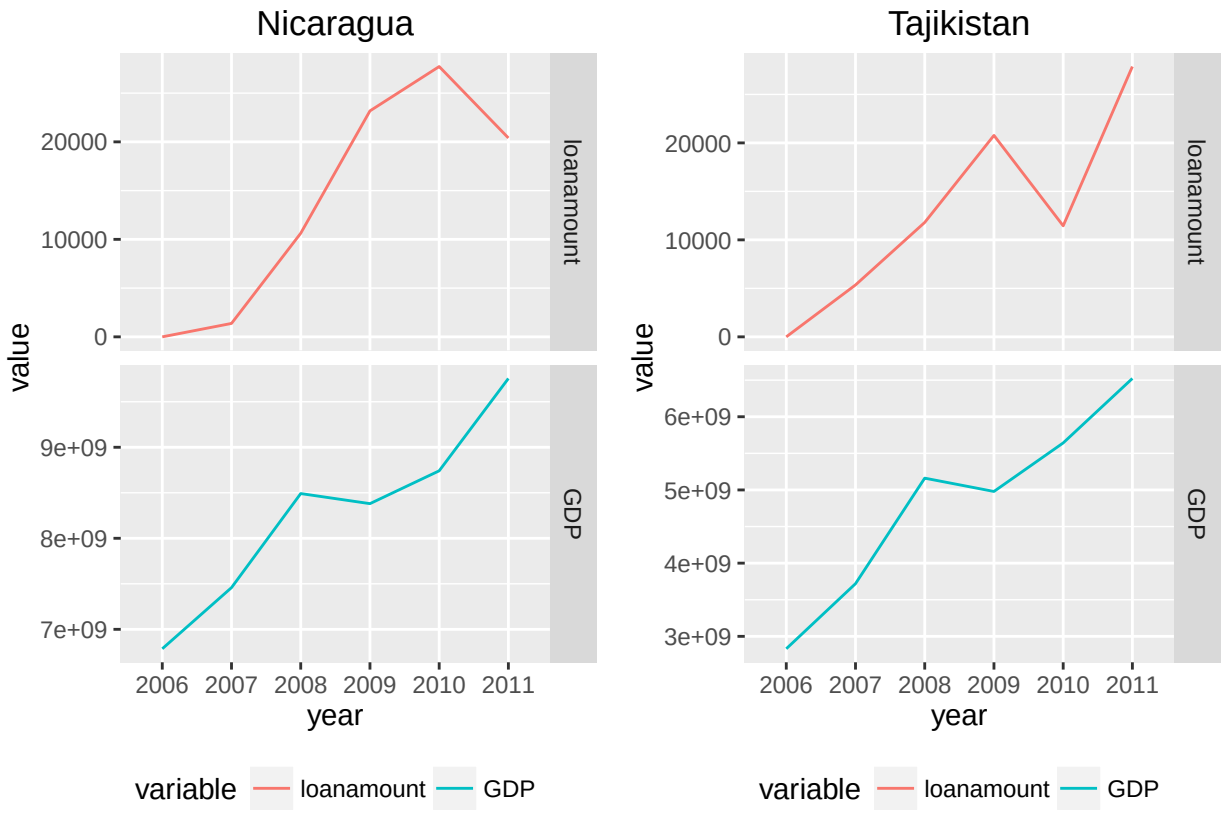
[Write takeaway]

```
grid.arrange(p5, p6, ncol=2)
```



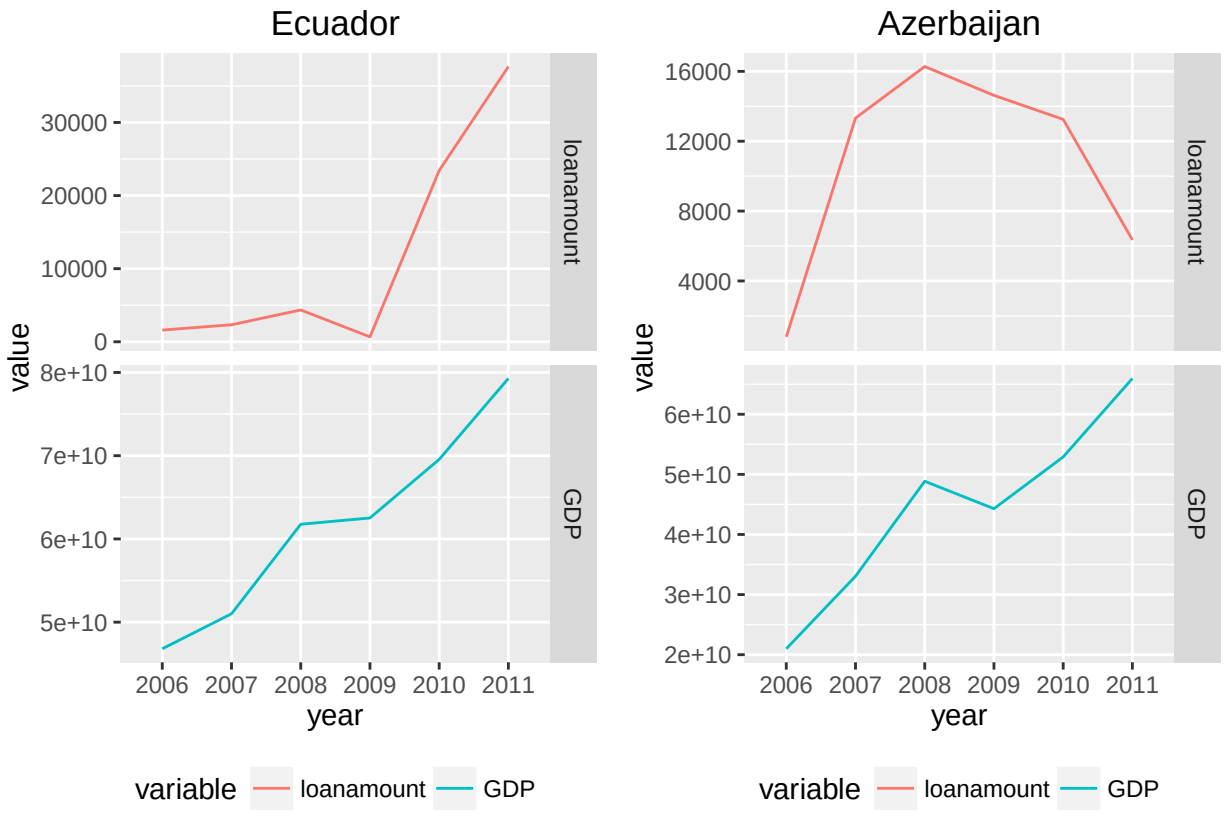
[Write takeaway]

```
grid.arrange(p7, p8, ncol=2)
```



[Write takeaway]

```
grid.arrange(p9, p10, ncol=2)
```



[Write takeaway]

Comparing the countries having the maximum loan requirement we observe that the countries with the lowest GDP are the ones asking for maximum loan amount.

4.7 Average loan amount, paid amount and sector

Kiva doesn't guarantee that loans will be paid back but its important for the lender to have that information before lending. We analyze average loan amount against the average paid back amount for every sector. `Loan amount`, `paid amount` and `sector` are required for this analysis. The average value of loan and paid amount for each sector is calculated and represented in bar graph.

Convert `paid_amount` from charactor to numeric. Create `sel_finaldf` data frame from `loan.df` by selecting `loan_amount` and `sector`.

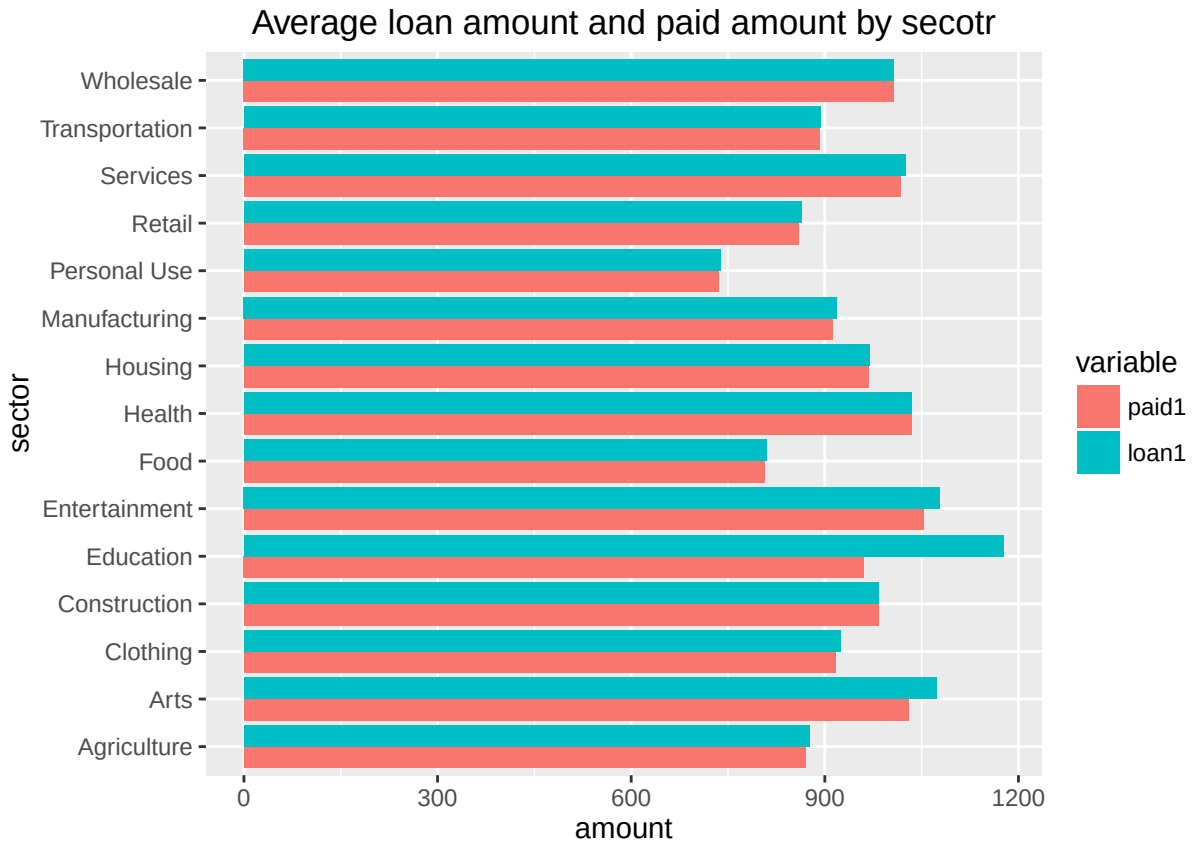
```
loan.df["paid_amount"] <- as.numeric(loan.df$paid_amount)

sel_finaldf <- select(loan.df,paid_amount,loan_amount,sector)
sel_finaldf <- na.omit(sel_finaldf)
```

We calculate the mean value of paid amount and loan amount for each sector.

```
sel_bysec <- ddply(sel_finaldf, .(sector), summarize,
                   paid1 = mean(paid_amount),
                   loan1 = mean(as.numeric(loan_amount)))
```

We calculate average loan amount and paid amount by sector.

```
#sel_bysec[["total_val"]] <- sel_bysec$paid1 + sel_bysec$loan1
dfm <- melt(sel_bysec, id.vars = c('sector'))
ggplot(dfm, aes(x = factor(sector), y = value, fill = variable) ) +
  geom_bar(stat="identity", position = 'dodge') +
  ggtitle("Average loan amount and paid amount by secotr") +
  xlab("sector") +
  ylab("amount") +
  coord_flip()
```



Its impressive to observe that except education sector almost all sectors maintain a good loan repayment capability. We also conclude that despite being an important sector for development, lenders might want to lend less money to education sector because of high number of loan defaulters.

4.8 Loan amount, Status and Sector

To understand the working of loans, status plays an imporant role. To analyze the status variable we create a factor variable called `statusF`, from the `status` character variable.

```
loan.df$statusF <- factor(loan.df$status)
table(loan.df$statusF)
```

```
##
##    defaulted      expired in_repayment         paid     refunded
##         1628          209          224        17737           67
```

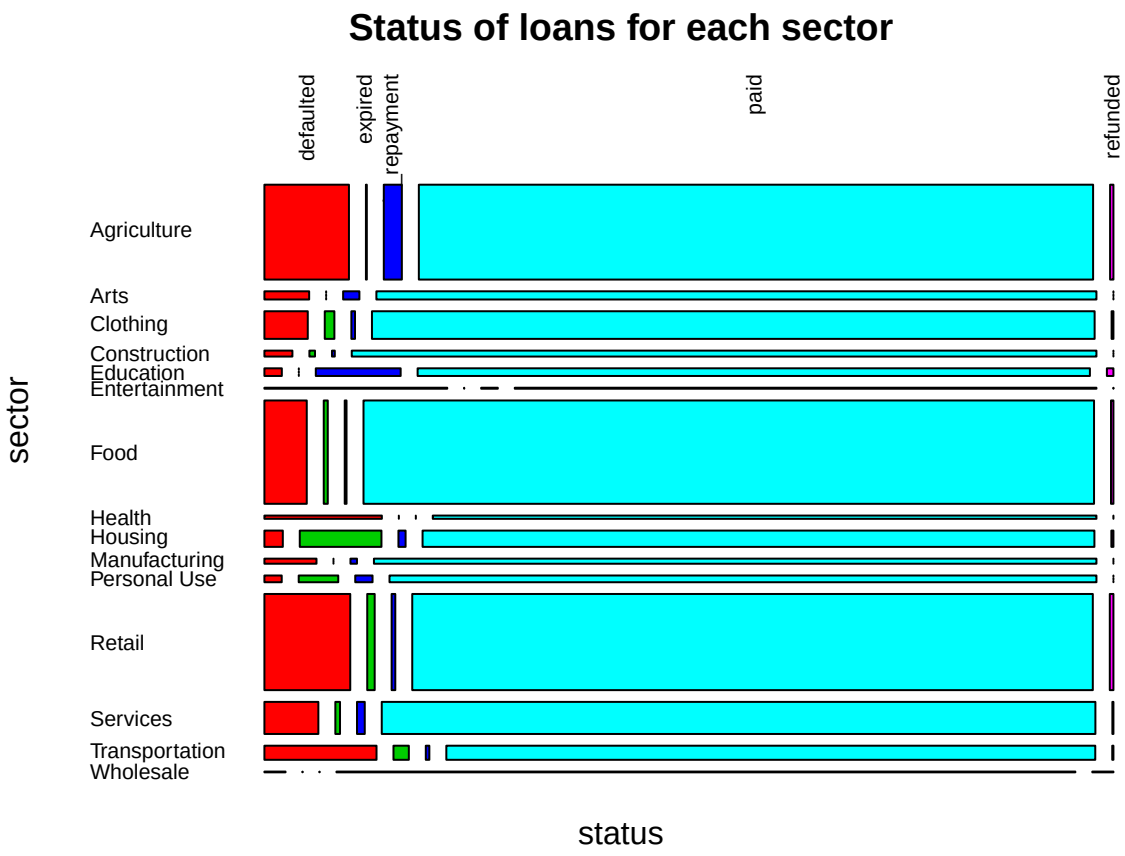
We see that there are 8 loan statuses which means:

- defaulted Loans are never paid back after not being paid for 6 months. They are financial loss to the lender of that loan.
- expired If the loan doesn't get fully funded within 30 days then it "expires".
- funded Loans that are completely funded are in "funded" state.
- fundraising These are the loans which are not yet funded. Lenders can only lend money to these loans.
- in_repayment When the loan is in repayment, it means that the loan has been disbursed to the borrower and they are in process of using the funds.
- paid These are the loans which are fully paid back by the borrower.
- refunded Few loans or a portion of it needs to be refunded, those loans maintain "refunded" status.

We observe that 77 percent of the loans have paid status and only 2 percent loans are default. This increases the confidence of lenders in Kiva's model.

We further visualise the status of loans based on sector.

```
op <- par(mar = rep(2, 4))
mosaicplot(sector_F ~ statusF, data = loan.df,
           main = "Status of loans for each sector",
           xlab = "status", ylab = "sector",
           dir = c('h','v'), las = 2,
           col = c(2:9), cex.axis = 0.7)
```



```
par(op)
```

We conclude that majority of the loans are paid back, however some borrowers within the retail and food sector default in repayment.

4.9 The time from funded date to payback date by top 10 loan amount countries

Kiva does not provide any information to lenders about the time period of when their loan will be repayed but it's imperative for a lender to know about the timeline before lending money. We analyze the time period when a loan is funded and is paid back for the countries which have the most loan requirement.

We calculate the difference in days by subtracting funded date from paid date and convert the days.

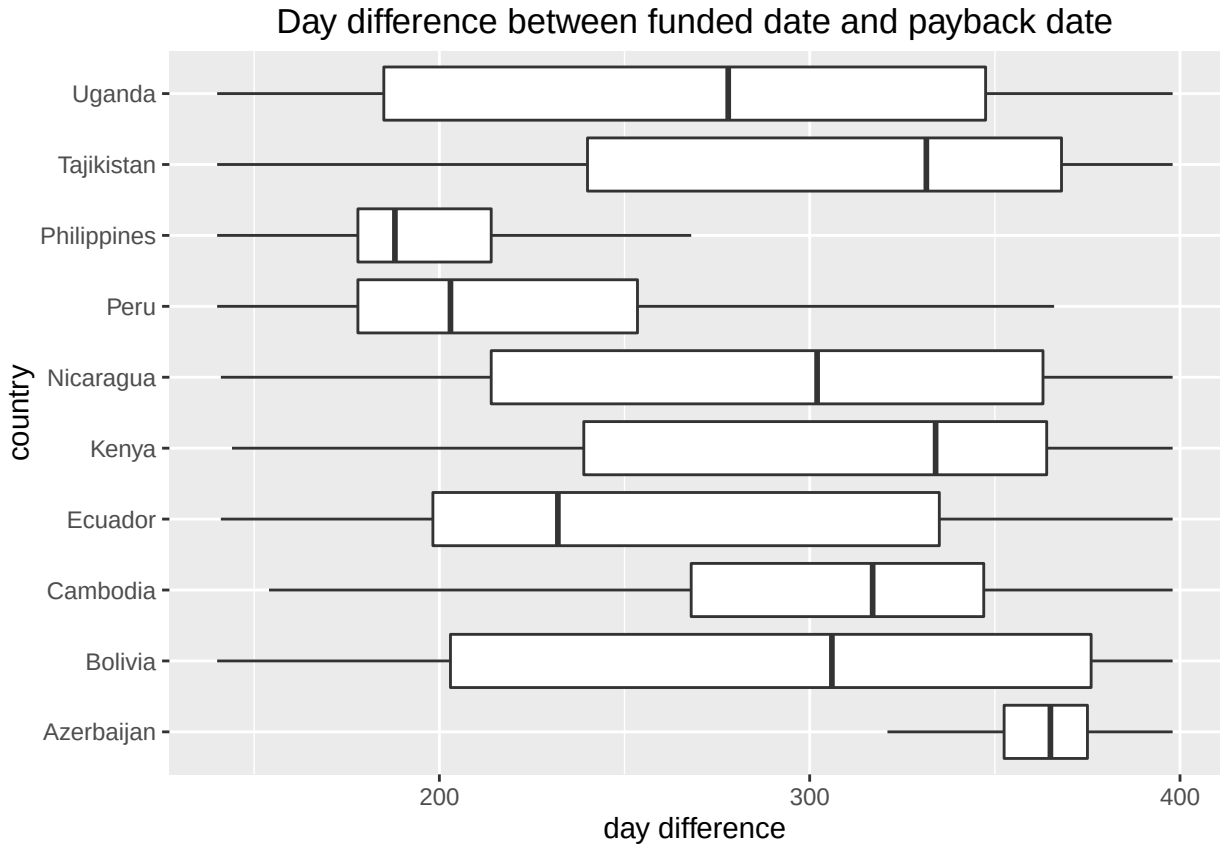
```
loan.df$day_diff <-
  as.numeric(loan.df$paid_date_cleaned - loan.df$funded_date_cleaned, "days")
```

We select top 10 country by amount using filter.

```
loan.df_date <-
  filter(loan.df, loan.df$countryF %in% c(top10_country_name))
```

Plot boxplot and change the scale of the graph to eliminate outliers.

```
loan.df_date %>%
  ggplot(aes(countryF, day_diff)) +
  ggtitle("Day difference between funded date and payback date") +
  xlab("country") +
  ylab("day difference") +
  geom_boxplot(outlier.shape = NA) +
  scale_y_continuous(limits = quantile(loan.df_date$day_diff, c(0.1, 0.9), na.rm = TRUE)) +
  coord_flip()
```



We observe that most of the loans are payed back within an year. We also conclude that Phillippines has the shortest average payback period and Kenya, Cambodia have highest average payback period among top 10 borrowers. Therefore lending money in Phillipines is the best option for lender.

5. Relationships between variables

The loan amount is compared with other variables such as `country`, `sector`, `activity`, `status`, `lendercount`. The average loan amount is calcualted and a new column is created, which takes the value 0 if the loan amount is less than average loan amount and value 1 if the loan amount is greater than average loan amount. This tells which particular sector or country or activity takes loan amount above average most times and which others takes less amount.

Below code converts `loan_amount` column into numeric value. A new dataframe is created using `dplyr` function. This dataframe has new column, `loan_amount_check` which has a value 0 for loan amounts below its average and has a value 1 for the loan amount higher than its average value. Another new column `lender_count_quant` is created which groups the lender count into 4 quantiles. Q1 denotes lender counts, which have values in between 0 and 25% of total counts, Q2 in between 25% and 50% of total counts, Q3 in between 50% and 75% of total counts and Q4 in between 75% and 100% of total lender counts. Finally, the variables that are required are selected using select function.

```
loan.df["loan_amount"] <- as.numeric(loan.df$loan_amount)
loan.df["lender_count"] <- as.numeric(loan.df$lender_count)
finaldf_loanamount <- loan.df %>%
  mutate(loan_amount_check = ifelse(loan_amount >=mean(loan_amount), 1, 0)) %>%
  mutate(lender_count_quant = cut(lender_count,
      quantile (lender_count, c (0, .25, .5, .75, 1)),
```

```
          labels=c('Q1','Q2','Q3','Q4'))) %>%
  select(lender_count_quant,status,activity,sector,
           location.country,loan_amount_check,borrowers.gender)

finaldf_loanamount <- data.frame(finaldf_loanamount)
```

For association rules to be applied, the variables should be factors. Hence the required variables are converted into factors.

```
finaldf_loanamount["lender_count_quant"] <- as.factor(finaldf_loanamount$lender_count_quant)
finaldf_loanamount["status"] <- as.factor(finaldf_loanamount$status)
finaldf_loanamount["activity"] <- as.factor(finaldf_loanamount$activity)
finaldf_loanamount["sector"] <- as.factor(finaldf_loanamount$sector)
finaldf_loanamount["location.country"] <- as.factor(finaldf_loanamount$location.country)
finaldf_loanamount["loan_amount_check"] <- as.factor(finaldf_loanamount$loan_amount_check)
finaldf_loanamount["borrowers.gender"]  <- as.factor(finaldf_loanamount$borrowers.gender)
```

For applying Apriori rule, parameters can be set to control the number of results. These are support of 0.01, confidence of 0.5 and maxlen of 2. Since our analysis focus on loan amount, rhs value is set to two different values of `loan_amount_check` (0 and 1) and lhs is set to default. The result is sorted by lift value.

```
apriori.appearance = list(rhs=c("loan_amount_check=0",
      "loan_amount_check=1"),default="lhs")
apriori.parameter = list(support=0.01,confidence=0.2,minlen=1,maxlen=3)
rules = apriori(finaldf_loanamount, parameter =
                  apriori.parameter,appearance =   apriori.appearance)
rules.sorted <- sort(rules, by = "lift")
```

The redundat rules are found and removed using the below code

```
subset.matrix <- is.subset(rules.sorted, rules.sorted)
subset.matrix[lower.tri(subset.matrix, diag=T)] <- NA
redundant <- colSums(subset.matrix, na.rm=T) >= 1

rules.pruned <- rules.sorted[!redundant]
inspect(rules.pruned)
```

```
##     lhs                                    rhs                      support confidence     lift
## 1   {lender_count_quant=Q4,
##      location.country=Paraguay}         => {loan_amount_check=1} 0.01379310  1.0000000 3.217525
## 2   {lender_count_quant=Q4,
##      location.country=Ecuador}          => {loan_amount_check=1} 0.01067204  1.0000000 3.217525
## 3   {lender_count_quant=Q4,
##      activity=Grocery Store}            => {loan_amount_check=1} 0.01047068  1.0000000 3.217525
## 4   {lender_count_quant=Q4,
##      location.country=Peru}             => {loan_amount_check=1} 0.01399446  1.0000000 3.217525
## 5   {lender_count_quant=Q4,
##      activity=Retail}                   => {loan_amount_check=1} 0.01867606  0.9973118 3.208876
## 6   {lender_count_quant=Q4,
##      sector=Retail}                     => {loan_amount_check=1} 0.04681601  0.9967846 3.207179
## 7   {lender_count_quant=Q4,
##      location.country=Mongolia}         => {loan_amount_check=1} 0.01414548  0.9964539 3.206115
```

```
## 8   {lender_count_quant=Q4,
##      activity=Clothing Sales}          => {loan_amount_check=1} 0.01339039  0.9962547 3.205474
## 9   {lender_count_quant=Q4,
##      activity=Agriculture}             => {loan_amount_check=1} 0.01213189  0.9958678 3.204230
## 10  {lender_count_quant=Q4,
##      activity=General Store}           => {loan_amount_check=1} 0.01052102  0.9952381 3.202204
## 11  {lender_count_quant=Q4,
##      sector=Clothing}                  => {loan_amount_check=1} 0.01731689  0.9942197 3.198927
## 12  {lender_count_quant=Q4,
##      borrowers.gender=M}               => {loan_amount_check=1} 0.08089605  0.9932015 3.195651
## 13  {lender_count_quant=Q4,
##      status=paid}                      => {loan_amount_check=1} 0.22305563  0.9921630 3.192309
## 14  {lender_count_quant=Q4,
##      sector=Food}                      => {loan_amount_check=1} 0.05008809  0.9920239 3.191862
## 15  {lender_count_quant=Q4}           => {loan_amount_check=1} 0.24258747  0.9919720 3.191695
## 16  {location.country=Paraguay,
##      borrowers.gender=F}               => {loan_amount_check=1} 0.01565568  0.9255952 2.978126
## 17  {status=paid,
##      location.country=Paraguay}        => {loan_amount_check=1} 0.01711553  0.8762887 2.819481
## 18  {location.country=Paraguay}       => {loan_amount_check=1} 0.01731689  0.8730964 2.809210
## 19  {location.country=Mongolia}       => {loan_amount_check=1} 0.01525296  0.7870130 2.532234
## 20  {location.country=Bolivia}        => {loan_amount_check=1} 0.01369242  0.6650367 2.139772
## 21  {activity=Livestock}              => {loan_amount_check=1} 0.01097408  0.6106443 1.964763
## 22  {location.country=Ecuador,
##      borrowers.gender=F}               => {loan_amount_check=1} 0.01031966  0.5662983 1.822079
## 23  {location.country=Ecuador}        => {loan_amount_check=1} 0.01500126  0.5518519 1.775597
## 24  {lender_count_quant=Q1,
##      location.country=Liberia}         => {loan_amount_check=0} 0.01031966  1.0000000 1.450953
## 25  {lender_count_quant=Q1,
##      location.country=Uganda}          => {loan_amount_check=0} 0.01570602  1.0000000 1.450953
## 26  {lender_count_quant=Q1,
##      location.country=Philippines}     => {loan_amount_check=0} 0.10692172  1.0000000 1.450953
## 27  {lender_count_quant=Q1,
##      location.country=Kenya}           => {loan_amount_check=0} 0.03946640  0.9987261 1.449105
## 28  {lender_count_quant=Q1,
##      location.country=Peru}            => {loan_amount_check=0} 0.01525296  0.9967105 1.446180
## 29  {lender_count_quant=Q1,
##      activity=General Store}           => {loan_amount_check=0} 0.04007048  0.9962453 1.445505
## 30  {lender_count_quant=Q1,
##      location.country=Ghana}           => {loan_amount_check=0} 0.01203121  0.9958333 1.444908
## 31  {lender_count_quant=Q2,
##      location.country=Philippines}     => {loan_amount_check=0} 0.04772212  0.9957983 1.444857
## 32  {lender_count_quant=Q1,
##      sector=Transportation}            => {loan_amount_check=0} 0.01157815  0.9956710 1.444672
## 33  {status=defaulted,
##      location.country=Philippines}     => {loan_amount_check=0} 0.03408004  0.9955882 1.444552
## 34  {activity=Motorcycle Transport,
##      location.country=Philippines}     => {loan_amount_check=0} 0.01031966  0.9951456 1.443910
## 35  {lender_count_quant=Q1,
##      status=defaulted}                 => {loan_amount_check=0} 0.03241883  0.9938272 1.441997
## 36  {lender_count_quant=Q1,
##      activity=Farming}                 => {loan_amount_check=0} 0.01892776  0.9920844 1.439468
## 37  {lender_count_quant=Q1,
##      location.country=Pakistan}        => {loan_amount_check=0} 0.01016864  0.9901961 1.436728
```

```
## 38  {lender_count_quant=Q1,
##      sector=Food}                          => {loan_amount_check=0} 0.08255726  0.9891435 1.435201
## 39  {lender_count_quant=Q1,
##      sector=Agriculture}                   => {loan_amount_check=0} 0.04872892  0.9887640 1.434650
## 40  {lender_count_quant=Q1,
##      status=paid}                          => {loan_amount_check=0} 0.23830858  0.9885153 1.434289
## 41  {sector=Transportation,
##      location.country=Philippines}         => {loan_amount_check=0} 0.01228291  0.9878543 1.433330
## 42  {lender_count_quant=Q2,
##      activity=Farming}                     => {loan_amount_check=0} 0.02008558  0.9876238 1.432996
## 43  {lender_count_quant=Q2,
##      activity=Food Production/Sales}       => {loan_amount_check=0} 0.01193053  0.9875000 1.432816
## 44  {lender_count_quant=Q2,
##      sector=Clothing}                      => {loan_amount_check=0} 0.01570602  0.9873418 1.432587
## 45  {lender_count_quant=Q1,
##      borrowers.gender=F}                   => {loan_amount_check=0} 0.22693179  0.9872974 1.432522
## 46  {sector=Services,
##      borrowers.gender=M}                   => {loan_amount_check=1} 0.01021898  0.4451754 1.432363
## 47  {lender_count_quant=Q1,
##      activity=Fruits & Vegetables}         => {loan_amount_check=0} 0.01097408  0.9864253 1.431257
## 48  {lender_count_quant=Q1,
##      sector=Retail}                        => {loan_amount_check=0} 0.07359678  0.9851752 1.429443
## 49  {lender_count_quant=Q2,
##      sector=Services}                      => {loan_amount_check=0} 0.01520262  0.9837134 1.427322
## 50  {lender_count_quant=Q1}                => {loan_amount_check=0} 0.27727158  0.9833958 1.426861
## 51  {lender_count_quant=Q2,
##      activity=Clothing Sales}              => {loan_amount_check=0} 0.01182985  0.9832636 1.426669
## 52  {lender_count_quant=Q2,
##      location.country=Kenya}               => {loan_amount_check=0} 0.02768689  0.9821429 1.425043
## 53  {lender_count_quant=Q2,
##      status=paid}                          => {loan_amount_check=0} 0.20427888  0.9811412 1.423590
## 54  {lender_count_quant=Q2,
##      location.country=Nicaragua}           => {loan_amount_check=0} 0.01268563  0.9805447 1.422725
## 55  {lender_count_quant=Q2,
##      sector=Food}                          => {loan_amount_check=0} 0.05517241  0.9803220 1.422401
## 56  {sector=Food,
##      location.country=Philippines}         => {loan_amount_check=0} 0.04455072  0.9800664 1.422031
## 57  {lender_count_quant=Q2,
##      borrowers.gender=F}                   => {loan_amount_check=0} 0.16778253  0.9788546 1.420272
## 58  {lender_count_quant=Q2,
##      location.country=Peru}                => {loan_amount_check=0} 0.01958218  0.9773869 1.418143
## 59  {lender_count_quant=Q2,
##      sector=Agriculture}                   => {loan_amount_check=0} 0.05174931  0.9771863 1.417852
## 60  {lender_count_quant=Q2,
##      activity=General Store}               => {loan_amount_check=0} 0.02527058  0.9766537 1.417079
## 61  {lender_count_quant=Q2,
##      status=defaulted}                     => {loan_amount_check=0} 0.01469922  0.9765886 1.416984
## 62  {activity=Pigs,
##      location.country=Philippines}         => {loan_amount_check=0} 0.01218223  0.9758065 1.415849
## 63  {lender_count_quant=Q2}                => {loan_amount_check=0} 0.22426378  0.9750492 1.414751
## 64  {location.country=Liberia,
##      borrowers.gender=F}                   => {loan_amount_check=0} 0.01228291  0.9644269 1.399338
## 65  {location.country=Philippines,
##      borrowers.gender=F}                   => {loan_amount_check=0} 0.15615404  0.9642524 1.399085
```

```
## 66  {status=defaulted,
##      activity=General Store}             => {loan_amount_check=0} 0.01676315  0.9624277 1.396438
## 67  {location.country=Philippines}       => {loan_amount_check=0} 0.17578656  0.9606602 1.393873
## 68  {sector=Food,
##      location.country=Ghana}             => {loan_amount_check=0} 0.01641077  0.9504373 1.379040
## 69  {sector=Food,
##      borrowers.gender=M}                 => {loan_amount_check=1} 0.01610873  0.4283802 1.378324
## 70  {activity=Fishing}                   => {loan_amount_check=0} 0.01122577  0.9489362 1.376862
## 71  {activity=Grocery Store}             => {loan_amount_check=1} 0.01344072  0.4244833 1.365786
## 72  {status=paid,
##      location.country=Ghana}             => {loan_amount_check=0} 0.02325698  0.9371197 1.359717
## 73  {location.country=Liberia}           => {loan_amount_check=0} 0.01273597  0.9370370 1.359597
## 74  {location.country=Ghana,
##      borrowers.gender=F}                 => {loan_amount_check=0} 0.02295495  0.9363450 1.358593
## 75  {location.country=Ghana}             => {loan_amount_check=0} 0.02436446  0.9343629 1.355717
## 76  {activity=Motorcycle Transport}      => {loan_amount_check=0} 0.01404480  0.9300000 1.349386
## 77  {sector=Food,
##      location.country=Kenya}             => {loan_amount_check=0} 0.01922980  0.9249395 1.342044
## 78  {status=paid,
##      location.country=Kenya}             => {loan_amount_check=0} 0.07500629  0.9057751 1.314237
## 79  {sector=Retail,
##      location.country=Kenya}             => {loan_amount_check=0} 0.01661213  0.9041096 1.311821
## 80  {activity=Fish Selling,
##      borrowers.gender=F}                 => {loan_amount_check=0} 0.01394412  0.9022801 1.309166
## 81  {location.country=Kenya,
##      borrowers.gender=F}                 => {loan_amount_check=0} 0.06267304  0.9002169 1.306173
## 82  {status=paid,
##      borrowers.gender=M}                 => {loan_amount_check=1} 0.09393405  0.4033722 1.297860
## 83  {status=defaulted,
##      sector=Retail}                      => {loan_amount_check=0} 0.02230053  0.8931452 1.295912
## 84  {activity=Fish Selling}              => {loan_amount_check=0} 0.01550466  0.8901734 1.291600
## 85  {borrowers.gender=M}                 => {loan_amount_check=1} 0.10621696  0.4011407 1.290680
## 86  {activity=Retail,
##      borrowers.gender=F}                 => {loan_amount_check=1} 0.01983388  0.3979798 1.280510
## 87  {location.country=Kenya}             => {loan_amount_check=0} 0.09468915  0.8806180 1.277735
## 88  {location.country=Tajikistan}        => {loan_amount_check=1} 0.01213189  0.3957307 1.273273
## 89  {sector=Food,
##      location.country=El Salvador}       => {loan_amount_check=0} 0.01042034  0.8734177 1.267288
## 90  {activity=General Store,
##      borrowers.gender=F}                 => {loan_amount_check=0} 0.07389882  0.8691533 1.261101
## 91  {location.country=Samoa,
##      borrowers.gender=F}                 => {loan_amount_check=0} 0.01233325  0.8626761 1.251703
## 92  {location.country=Samoa}             => {loan_amount_check=0} 0.01243393  0.8606272 1.248730
## 93  {activity=Personal Housing Expenses} => {loan_amount_check=1} 0.01374276  0.3855932 1.240656
## 94  {activity=General Store}             => {loan_amount_check=0} 0.08240624  0.8526042 1.237089
## 95  {sector=Housing}                     => {loan_amount_check=1} 0.01479990  0.3843137 1.236539
## 96  {status=paid,
##      activity=Retail}                    => {loan_amount_check=1} 0.02174679  0.3799472 1.222490
## 97  {activity=Food Market,
##      borrowers.gender=F}                 => {loan_amount_check=0} 0.01716587  0.8399015 1.218658
## 98  {activity=Retail}                    => {loan_amount_check=1} 0.02310597  0.3784007 1.217514
## 99  {location.country=El Salvador,
##      borrowers.gender=F}                 => {loan_amount_check=0} 0.01973320  0.8287526 1.202481
## 100 {status=paid,
```

```
##      location.country=El Salvador}       => {loan_amount_check=0} 0.02990184  0.8284519 1.202045
## 101 {status=defaulted,
##      sector=Food}                        => {loan_amount_check=0} 0.01092374  0.8250951 1.197174
## 102 {lender_count_quant=Q3,
##      location.country=Peru}              => {loan_amount_check=0} 0.01540398  0.8247978 1.196743
## 103 {activity=Fruits & Vegetables,
##      borrowers.gender=F}                 => {loan_amount_check=0} 0.01988422  0.8212058 1.191531
## 104 {sector=Services}                    => {loan_amount_check=1} 0.02788825  0.3678619 1.183605
## 105 {location.country=El Salvador}       => {loan_amount_check=0} 0.03126101  0.8128272 1.179374
## 106 {sector=Agriculture,
##      location.country=Peru}              => {loan_amount_check=0} 0.01117543  0.8102190 1.175590
## 107 {status=paid,
##      location.country=Pakistan}          => {loan_amount_check=0} 0.01359174  0.8083832 1.172926
## 108 {activity=Pigs,
##      borrowers.gender=F}                 => {loan_amount_check=0} 0.01414548  0.8028571 1.164908
## 109 {activity=Food Production/Sales,
##      borrowers.gender=F}                 => {loan_amount_check=0} 0.03533854  0.8022857 1.164079
## 110 {location.country=Nicaragua,
##      borrowers.gender=M}                 => {loan_amount_check=0} 0.01031966  0.7976654 1.157375
## 111 {location.country=Uganda,
##      borrowers.gender=F}                 => {loan_amount_check=0} 0.02184747  0.7963303 1.155438
## 112 {status=paid,
##      activity=Food Market}               => {loan_amount_check=0} 0.01746791  0.7958716 1.154772
## 113 {activity=Pigs}                      => {loan_amount_check=0} 0.01615907  0.7945545 1.152861
## 114 {status=paid,
##      location.country=Nicaragua}         => {loan_amount_check=0} 0.02959980  0.7913863 1.148264
## 115 {activity=Agriculture}               => {loan_amount_check=1} 0.01555500  0.3564014 1.146730
## 116 {lender_count_quant=Q3,
##      sector=Food}                        => {loan_amount_check=0} 0.04268814  0.7903075 1.146699
## 117 {activity=Food Production/Sales}     => {loan_amount_check=0} 0.03710043  0.7882353 1.143693
## 118 {lender_count_quant=Q3,
##      borrowers.gender=F}                 => {loan_amount_check=0} 0.13450793  0.7875037 1.142631
## 119 {lender_count_quant=Q3,
##      sector=Retail}                      => {loan_amount_check=0} 0.03971810  0.7842942 1.137974
## 120 {activity=Food Market}               => {loan_amount_check=0} 0.01827335  0.7840173 1.137572
## 121 {location.country=Nicaragua}         => {loan_amount_check=0} 0.03085829  0.7838875 1.137384
## 122 {lender_count_quant=Q3,
##      activity=Retail}                    => {loan_amount_check=0} 0.01127611  0.7832168 1.136411
## 123 {lender_count_quant=Q3,
##      activity=Farming}                   => {loan_amount_check=0} 0.02139441  0.7826888 1.135645
## 124 {activity=Fruits & Vegetables}       => {loan_amount_check=0} 0.02315631  0.7809847 1.133172
## 125 {sector=Transportation,
##      borrowers.gender=F}                 => {loan_amount_check=0} 0.01057136  0.7806691 1.132714
## 126 {location.country=Peru,
##      borrowers.gender=M}                 => {loan_amount_check=0} 0.01162849  0.7804054 1.132332
## 127 {status=defaulted,
##      borrowers.gender=F}                 => {loan_amount_check=0} 0.05023911  0.7796875 1.131290
## 128 {status=paid,
##      sector=Agriculture}                 => {loan_amount_check=1} 0.06795872  0.3511053 1.129690
## 129 {activity=Clothing Sales}            => {loan_amount_check=1} 0.01726655  0.3503575 1.127284
## 130 {lender_count_quant=Q3,
##      status=paid}                        => {loan_amount_check=0} 0.16919205  0.7733548 1.122102
## 131 {sector=Food,
##      borrowers.gender=F}                 => {loan_amount_check=0} 0.15932545  0.7710110 1.118701
```

```
## 132 {status=defaulted,
##      activity=Farming}                  => {loan_amount_check=0} 0.01369242  0.7705382 1.118015
## 133 {status=defaulted}                  => {loan_amount_check=0} 0.06282406  0.7665848 1.112279
## 134 {lender_count_quant=Q3}             => {loan_amount_check=0} 0.18570350  0.7629783 1.107046
## 135 {location.country=Uganda}           => {loan_amount_check=0} 0.03332494  0.7583047 1.100265
## 136 {activity=Farming,
##      borrowers.gender=F}                => {loan_amount_check=0} 0.04243645  0.7574124 1.098970
## 137 {status=paid,
##      sector=Personal Use}               => {loan_amount_check=0} 0.01082306  0.7517483 1.090752
## 138 {sector=Retail,
##      borrowers.gender=F}                => {loan_amount_check=0} 0.14034734  0.7496639 1.087727
## 139 {sector=Clothing}                   => {loan_amount_check=1} 0.02204883  0.3377024 1.086566
## 140 {activity=Farming,
##      location.country=Cambodia}         => {loan_amount_check=0} 0.01031966  0.7481752 1.085567
## 141 {sector=Agriculture}                => {loan_amount_check=1} 0.07550969  0.3367759 1.083585
## 142 {status=paid,
##      location.country=Peru}             => {loan_amount_check=0} 0.04923232  0.7420334 1.076656
## 143 {sector=Food}                       => {loan_amount_check=0} 0.18082054  0.7403133 1.074160
## 144 {sector=Agriculture,
##      location.country=Cambodia}         => {loan_amount_check=0} 0.01384344  0.7392473 1.072613
## 145 {location.country=Peru}             => {loan_amount_check=0} 0.05023911  0.7387121 1.071837
## 146 {sector=Retail}                     => {loan_amount_check=0} 0.16632268  0.7319451 1.062018
## 147 {sector=Personal Use}               => {loan_amount_check=0} 0.01162849  0.7310127 1.060665
## 148 {activity=Farming}                  => {loan_amount_check=0} 0.06086081  0.7256903 1.052943
## 149 {borrowers.gender=F}                => {loan_amount_check=0} 0.53063176  0.7217391 1.047210
## 150 {location.country=Vietnam}          => {loan_amount_check=0} 0.01052102  0.7206897 1.045687
## 151 {status=paid,
##      activity=Food Stall}               => {loan_amount_check=0} 0.01288699  0.7111111 1.031789
## 152 {activity=Food Stall}               => {loan_amount_check=0} 0.01349106  0.7089947 1.028718
## 153 {status=paid,
##      location.country=Cambodia}         => {loan_amount_check=0} 0.02265291  0.7042254 1.021798
## 154 {location.country=Cambodia}         => {loan_amount_check=0} 0.02265291  0.7031250 1.020201
## 155 {status=paid,
##      activity=Food}                     => {loan_amount_check=0} 0.01067204  0.6996700 1.015188
## 156 {activity=Food}                     => {loan_amount_check=0} 0.01162849  0.6978852 1.012599
## 157 {status=paid,
##      activity=Sewing}                   => {loan_amount_check=0} 0.01152781  0.6960486 1.009934
## 158 {status=paid}                       => {loan_amount_check=1} 0.27933551  0.3128488 1.006599
## 159 {sector=Transportation}             => {loan_amount_check=0} 0.02305563  0.6918429 1.003832
## 160 {}                                  => {loan_amount_check=1} 0.31079789  0.3107979 1.000000
## 161 {}                                  => {loan_amount_check=0} 0.68920211  0.6892021 1.000000
```

Based on sorting by lift value, the lenders in quantile 4 are associated with higher loan amounts in Bolivia. It is a known fact that large number of lenders are required if the loan amount is high and this rule also focuses on country apart from lender count.

```
inspect(rules.pruned[1])
```

```
##   lhs                              rhs                      support confidence     lift
## 1 {lender_count_quant=Q4,
##    location.country=Paraguay} => {loan_amount_check=1} 0.0137931          1 3.217525
```

This has a support value of 0.011% which indicates the percentage of the higher loans associated with large count of lenders and as well as in country Bolivia. It has a confidence of 96% and higher lift value 3.012.

Below result shows that large number of lenders are interested in services sector for lending higher loan amounts.

```
inspect(rules.pruned[4])
```

```
##   lhs                          rhs                       support confidence     lift
## 1 {lender_count_quant=Q4,
##    location.country=Peru} => {loan_amount_check=1} 0.01399446          1 3.217525
```

This has the fourth higher lift value of 2.58. The support value is 0.01 and the confidence value shows that among all sectors, lenders are interested in services sector and its 97% associated with higher loan amounts.

An interesting rule which we came across genders with respect to loans are provided below. Though the percentage of loans taken by males are low across various countries/sectors, the loan amount taken by them are higher.

```
inspect(rules.pruned[69])
```

```
##   lhs                      rhs                       support confidence     lift
## 1 {sector=Food,
##    borrowers.gender=M} => {loan_amount_check=1} 0.01610873  0.4283802 1.378324
```

This rule has a support value of 10% stating that 10% of males across various countries and sectors are associated with higher loan amount The confidence value shows that 41% of gender who are male are associated with higher loan amount. It has a lift value of 1.28 which indicates that it is 1.28 times more likely to occur in the data.

Another interesting rule with respect to gender and loan amount is explained below. We have already stated that females are involved in larger percentage of loans when compared to male. But they are associated with less amount of loans.

```
inspect(rules.pruned[136])
```

```
##   lhs                      rhs                       support confidence    lift
## 1 {activity=Farming,
##    borrowers.gender=F} => {loan_amount_check=0} 0.04243645  0.7574124 1.09897
```

This rule has very high support value of 52% stating that 52% of females across various countries and sectors are associated with smaller loan amount. The confidence value shows that 70% of gender who are female are associated with lesser loan amount. It has a lift value of 1.04 which indicates that it is 1.04 times more likely to occur in the data.

From the below result, we infer that people who have paid the loan amount are those who got loans less than average loan amount. Those who got higher loans are in other status.

```
inspect(rules.pruned[143])
```

```
##    lhs              rhs                     support   confidence lift
## 69 {sector=Food} => {loan_amount_check=0} 0.1808205 0.7403133  1.07416
```

This has higher support value of 53% and a confidence of 69% stating that of all the statuses that are paid, loan amount is of lesser value. A lift of 1.02 states that this scenario is 1.02 times more likely to happen with the data.

Relationship between status and other variables

The next analysis is regarding status of loans. We picked defaulted and in_repayment statuses which are the loans that are not paid back completely. We wanted to see how much risk is associated with higher loan amounts and hence filtered loans above the average value.

A new dataframe is created using the below dplyr function which filters defaulted and in_repayment status and selects the required variable for comparison.

```
avgloan_df <- loan.df[loan.df$loan_amount > mean(loan.df$loan_amount),]
finaldf_status <- avgloan_df %>%
  select(status,sector,
           location.country,borrowers.gender) %>%
  filter(status == "defaulted" | status == "in_repayment")

finaldf_status <- data.frame(finaldf_status)
```

For association rules to be applied, the variables should be factors. Hence the required variables are converted into factors.

```
finaldf_status["status"] <- as.factor(finaldf_status$status)
finaldf_status["sector"] <- as.factor(finaldf_status$sector)
finaldf_status["location.country"] <- as.factor(finaldf_status$location.country)
finaldf_status["borrowers.gender"] <- as.factor(finaldf_status$borrowers.gender)
```

Apriori rule is applied by setting appropriate values to support, confidence and maxlen. Here we need to set confidence and support values to low (0.01) as only one status is getting predicted for higher values. Here defaulted and in_repayment status are given as rhs and rest as lhs values. The result is sorted by lift value.

```
apriori.appearance = list(rhs=c("status=defaulted","status=in_repayment"),
    default="lhs")
apriori.parameter = list(support=0.01,confidence=0.01,minlen=1,maxlen=3)

rules_status = apriori(finaldf_status, parameter =
                         apriori.parameter,appearance =   apriori.appearance)

rules_status.sorted <- sort(rules_status, by = "lift")
inspect(rules_status.sorted)
```

The redundant rules are found and removed using the below code

```
subset.matrix <- is.subset(rules_status.sorted, rules_status.sorted)
subset.matrix[lower.tri(subset.matrix, diag=T)] <- NA
redundant <- colSums(subset.matrix, na.rm=T) >= 1

rules_status.pruned <- rules_status.sorted[!redundant]
inspect(rules_status.pruned)
```

```
##    lhs                                rhs                       support confidence     lift
## 1  {location.country=Mongolia}     => {status=in_repayment} 0.02173913  1.0000000 5.750000
## 2  {location.country=India}        => {status=in_repayment} 0.06086957  1.0000000 5.750000
## 3  {location.country=Burkina Faso} => {status=in_repayment} 0.06086957  1.0000000 5.750000
```

```
## 4  {sector=Education,
##     borrowers.gender=M}           => {status=in_repayment} 0.05869565  0.9642857 5.544643
## 5  {sector=Education}             => {status=in_repayment} 0.06956522  0.8648649 4.972973
## 6  {sector=Housing}               => {status=in_repayment} 0.01521739  0.5000000 2.875000
## 7  {sector=Arts}                  => {status=in_repayment} 0.01521739  0.3684211 2.118421
## 8  {borrowers.gender=M}           => {status=in_repayment} 0.08260870  0.2794118 1.606618
## 9  {location.country=Albania}     => {status=defaulted}    0.01086957  1.0000000 1.210526
## 10 {location.country=Ecuador}     => {status=defaulted}    0.01086957  1.0000000 1.210526
## 11 {location.country=Indonesia}   => {status=defaulted}    0.01086957  1.0000000 1.210526
## 12 {location.country=Bolivia}     => {status=defaulted}    0.01086957  1.0000000 1.210526
## 13 {location.country=Togo}        => {status=defaulted}    0.01304348  1.0000000 1.210526
## 14 {location.country=Guatemala}   => {status=defaulted}    0.01739130  1.0000000 1.210526
## 15 {location.country=Peru}        => {status=defaulted}    0.02173913  1.0000000 1.210526
## 16 {sector=Health}                => {status=defaulted}    0.03695652  1.0000000 1.210526
## 17 {location.country=Rwanda}      => {status=defaulted}    0.05869565  1.0000000 1.210526
## 18 {location.country=Kenya}       => {status=defaulted}    0.21304348  1.0000000 1.210526
## 19 {location.country=Pakistan}    => {status=defaulted}    0.26956522  1.0000000 1.210526
## 20 {sector=Food,
##     location.country=Sierra Leone} => {status=defaulted}    0.01086957  1.0000000 1.210526
## 21 {sector=Retail,
##     location.country=Sierra Leone} => {status=defaulted}    0.01521739  1.0000000 1.210526
## 22 {location.country=Sierra Leone,
##     borrowers.gender=F}           => {status=defaulted}    0.02608696  1.0000000 1.210526
## 23 {sector=Food,
##     borrowers.gender=M}           => {status=defaulted}    0.01304348  1.0000000 1.210526
## 24 {sector=Retail,
##     borrowers.gender=M}           => {status=defaulted}    0.02826087  1.0000000 1.210526
## 25 {sector=Agriculture,
##     borrowers.gender=M}           => {status=defaulted}    0.08913043  0.9761905 1.181704
## 26 {sector=Agriculture}           => {status=defaulted}    0.28478261  0.9492754 1.149123
## 27 {location.country=Sierra Leone} => {status=defaulted}   0.03695652  0.9444444 1.143275
## 28 {sector=Services,
##     borrowers.gender=F}           => {status=defaulted}    0.08695652  0.9302326 1.126071
## 29 {sector=Services}              => {status=defaulted}    0.10869565  0.9259259 1.120858
## 30 {sector=Transportation,
##     borrowers.gender=F}           => {status=defaulted}    0.02608696  0.9230769 1.117409
## 31 {sector=Food}                  => {status=defaulted}    0.10000000  0.9200000 1.113684
## 32 {sector=Clothing,
##     borrowers.gender=F}           => {status=in_repayment} 0.01304348  0.1935484 1.112903
## 33 {sector=Retail}                => {status=defaulted}    0.11521739  0.8983051 1.087422
## 34 {location.country=Israel}      => {status=defaulted}    0.01739130  0.8888889 1.076023
## 35 {sector=Manufacturing}         => {status=defaulted}    0.01739130  0.8888889 1.076023
## 36 {sector=Transportation}        => {status=defaulted}    0.03478261  0.8888889 1.076023
## 37 {borrowers.gender=F}           => {status=defaulted}    0.61304348  0.8703704 1.053606
## 38 {sector=Clothing}              => {status=in_repayment} 0.01304348  0.1764706 1.014706
## 39 {location.country=Nicaragua}   => {status=defaulted}    0.01086957  0.8333333 1.008772
## 40 {}                             => {status=in_repayment} 0.17391304  0.1739130 1.000000
## 41 {}                             => {status=defaulted}    0.82608696  0.8260870 1.000000
```

After sorting the rules based on higher lift value, Togo is top country which has female genders associated with defaulted loan status.

```
inspect(rules_status.pruned[1])
```

```
##    lhs                             rhs                    support
## 14 {location.country=Mongolia} => {status=in_repayment} 0.02173913
##    confidence lift
## 14 1          5.75
```

They have a support value of 0.01 and confidence of 97% which determines the percentage of female genders who are associated with defaulted loan status. They have a lift value of 7.46

Similar to Togo, Pakistan is one other country where female genders are associated with defaulted loan status. It ranks next to Togo based on higher lift value.

```
inspect(rules_status.pruned[3])
```

```
##   lhs                                rhs                    support confidence lift
## 1 {location.country=Burkina Faso} => {status=in_repayment} 0.06086957          1 5.75
```

They have a support value of 0.012 and confidence of 63% which determines the percentage of female genders who are associated with defaulted loan status. They have a lift value of 4.90.

Another interesting rule which we found with respect to defaulted loan status is provided below.

```
inspect(rules_status.pruned[10])
```

```
##   lhs                            rhs                 support    confidence
## 4 {location.country=Ecuador} => {status=defaulted} 0.01086957 1
##   lift
## 4 1.210526
```

Agriculture is one sector where loans borrowed by males are in defaulted status. It has a support value of 0.013 and a confidence of 15% which states the percentage of agriculture sector in association with defaulted loan status. They have a lift value of 1.162.

Below result shows the percentage of male population associated with defaulted status.

```
inspect(rules_status.pruned[37])
```

```
##    lhs                      rhs                 support   confidence
## 40 {borrowers.gender=F} => {status=defaulted} 0.6130435 0.8703704
##    lift
## 40 1.053606
```

This rule has a support of 0.04%. The confidence value shows that 14% of gender who are male are associated with defaulted loan status. A lift of 1.12 states that this scenario is 1.12 times more likely to happen with the data.

From above results, it is clear that lenders should be more cautious while providing loans to countries such as Togo, Pakistan and also for agriculture sector as they are more likely to end up with defaulted loaan status.

Countries such as Vietnam, Nigeria and india have higher lift values with repect to `in_repayment` status.

Below rule shows the female association with respect to `in_repayment` status.

```
#inspect(rules_status.pruned[65])
```

This rule has very high support value of 60% stating that 60% of females across various countries and sectors have their loan status as `in_repayment`. The confidence value shows that 87% of gender who are female are associated with `in_repayment` loan status. It has a lift value of 1.008 which indicates that it is 1.008 times more likely to occur in the data.

As female population are trying to pay back the loan, there are two possibilities that can occur, either they can completely pay back the loan or they can be in defaulted loan status. As we have already stated that majority of female populations are associated with lower value of loan amounts, there are less chances that female population can be in defaulted status when it comes to larger value of loan amounts, even though they are associated with certain defaulted status.

6. Conclusion