Reading data into R for analysis :

CafeGreat <- read.csv("CafeGreatTransaction.csv")

Data Cleaning and Summary :

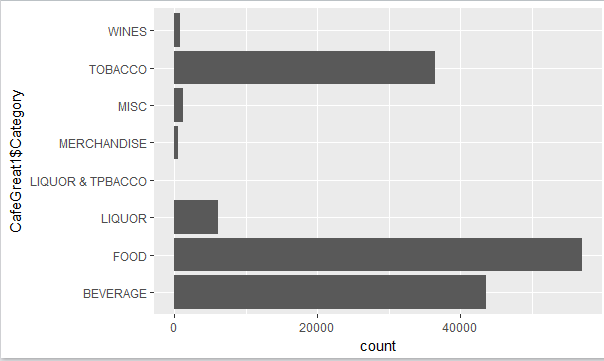
CafeGreat1<- CafeGreat[,-c(11)]

CafeGreat1$Category <- trimws(CafeGreat1$Category, "r")

CafeGreat1$Item.Desc <- trimws(CafeGreat1$Item.Desc, "r")

CafeGreat1$Item.Desc <- trimws(CafeGreat1$Item.Desc, "l")

ggplot(CafeGreat1, aes(CafeGreat1$Category)) + geom\_bar() + coord\_flip()



> summary(CafeGreat)

Date Bill.Number

31-Dec-10: 834 G0490530: 23

3-Apr-10 : 631 G0518006: 23

18-Dec-10: 620 G0489943: 21

29-Jan-11: 607 G0526679: 19

5-Feb-11 : 573 G0495644: 18

11-Dec-10: 572 G0489210: 17

(Other) :141993 (Other) :145709

Item.Desc Time

NIRVANA HOOKAH SINGLE : 8553 22:25:36: 33

MINT FLAVOUR SINGLE : 5817 23:35:33: 30

CAPPUCCINO : 5495 22:58:37: 29

GREAT LAKES SHAKE : 4895 23:02:58: 26

SAMBUCA : 4425 01:25:14: 25

POUTINE WITH FRIES : 3464 21:07:44: 25

(Other) :113181 (Other) :145662

Quantity Rate Tax Discount Total

Min. : 1.0 Min. : 0 Min. : 0 Min. : 0 Min. : 0

1st Qu.: 1.0 1st Qu.: 95 1st Qu.: 23 1st Qu.: 0 1st Qu.: 118

Median : 1.0 Median : 125 Median : 32 Median : 0 Median : 167

Mean : 1.1 Mean : 162 Mean : 49 Mean : 0 Mean : 225

3rd Qu.: 1.0 3rd Qu.: 225 3rd Qu.: 72 3rd Qu.: 0 3rd Qu.: 315

Max. :30.0 Max. :2100 Max. :2731 Max. :825 Max. :14231

Category X

Length:145830 Mode:logical

Class :character NA's:145830

Mode :character

> CafeGreat1<- CafeGreat[c(1,4,9)]

> View(CafeGreat1)

> library(lubridate)

> CafeGreat1$ParsedTime <- hms(as.character(CafeGreat1$Time))

> View(CafeGreat1)

> CafeGreat1$Hour <- hour(CafeGreat1$ParsedTime)

> View(CafeGreat1)

> CafeGreatNew <- CafeGreat1[c(1,5,3)]

> View(CafeGreatNew)

> CafeGreatNewGroup<-aggregate(CafeGreatNew$Total, by=list(Category=CafeGreatNew$Date,CafeGreatNew$Hour), FUN=sum)

> colnames(CafeGreatNewGroup) <- c("Date","Hour","TotalSales")

> View(CafeGreatNewGroup)

## Average Monthly Sales

> sum(CafeGreatNewGroup$TotalSales)/12

[1] 2733825

## Average Weekly Sales

> sum(CafeGreatNewGroup$TotalSales)/52

[1] 630883

## Average Daily Sales

> sum(CafeGreatNewGroup$TotalSales)/365

[1] 89879

## Total Sales

> sum(CafeGreatNewGroup$TotalSales)

[1] 3.3e+07

> View(CafeGreatSalesAggQ1)

> sum(CafeGreatSalesAggQ1$x)

[1] 163519

## Daily Sales

> SalesByDay<-aggregate(CafeGreatNewGroup$TotalSales, by=list(Category=CafeGreatNewGroup$Date), FUN=sum)

> View(SalesByDay)

## Monthly Sales

> CafeGreat1$TimeCons<-as.POSIXct(CafeGreat$Date,format="%d-%b-%y")

> CafeGreat1$Month <- month(CafeGreat1$TimeCons)

> View(CafeGreat1)

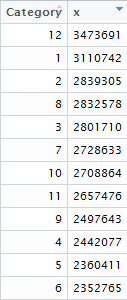
> CafeGreatMonthGrp<-CafeGreat1[c(6,7,3)]

> CafeGreatMonthGrpA<-aggregate(CafeGreatMonthGrp$Total, by=list(Category=CafeGreatMonthGrp$TimeCons,CafeGreatMonthGrp$Month), FUN=sum)

> View(CafeGreatMonthGrpA)

> SalesByMonth <- aggregate(CafeGreatMonthGrpA$x, by=list(Category=CafeGreatMonthGrpA$Group.2), FUN=sum)

> View(SalesByMonth)



## Top 20 Item sales by Quantity

> CafeGreatSales <- CafeGreat[c(1,3,5,9)]

> CafeGreatSalesAggQ <- aggregate(CafeGreatSales$Quantity, by=list(Category=CafeGreatSales$Date,CafeGreatSales$Item.Desc), FUN=sum)

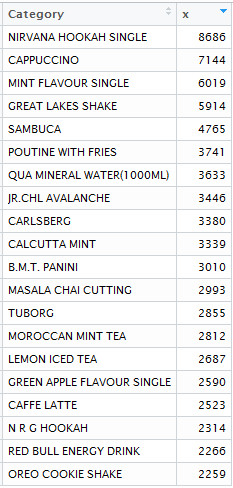
> CafeGreatSalesAggQ1 <- aggregate(CafeGreatSales$Quantity, by=list(Category=CafeGreatSales$Item.Desc), FUN=sum)

> library(dplyr)

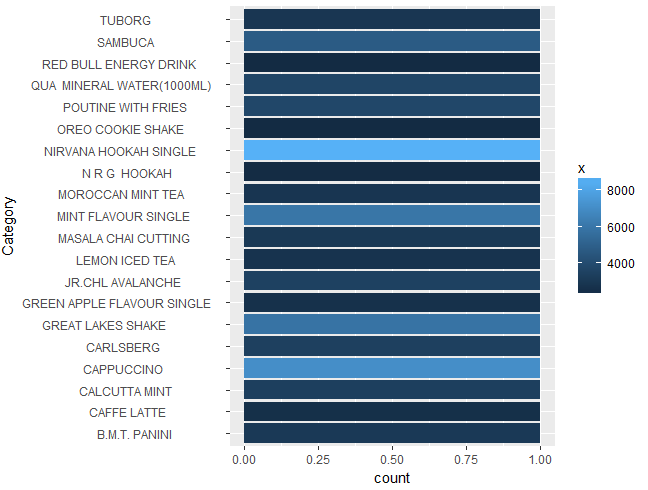
> temp<-CafeGreatSalesAggQ1 %>%

+ top\_n(20)

> View(temp)



ggplot(temp, aes(Category,fill=x)) + geom\_bar()+coord\_flip()



## Bottom 20 Item sales by Quantity

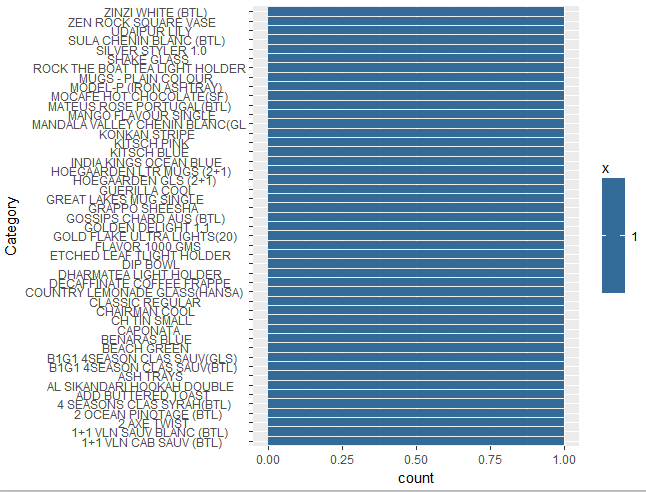
> temp1 <- CafeGreatSalesAggQ1 %>%

+ top\_n(-20)

> View(temp1)



> ggplot(temp1, aes(Category,fill=x)) + geom\_bar()+coord\_flip()



## Top 20 Item sales by Revenue

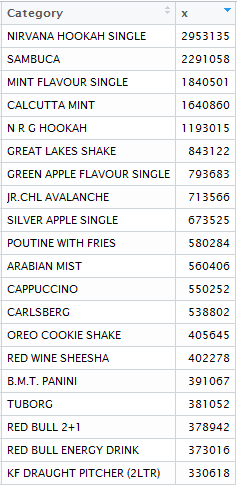
> CafeGreatSalesAggSales <- aggregate(CafeGreatSales$Total, by=list(Category=CafeGreatSales$Item.Desc), FUN=sum)

> View(CafeGreatSalesAggSales)

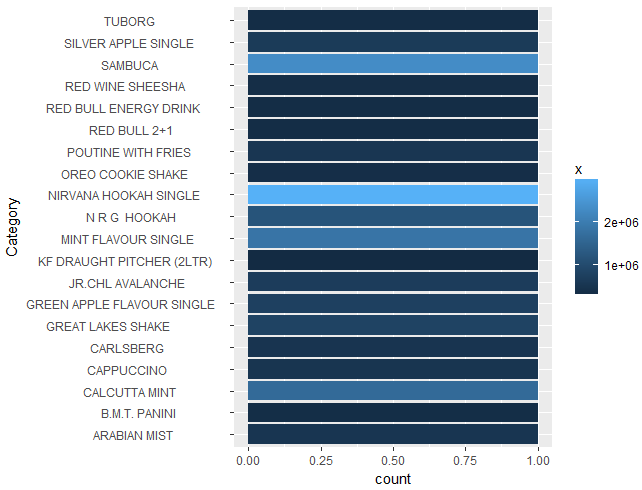
> temp2<-CafeGreatSalesAggSales %>%

+ top\_n(20)

> View(temp2)



> ggplot(temp2, aes(Category,fill=x)) + geom\_bar()+coord\_flip()

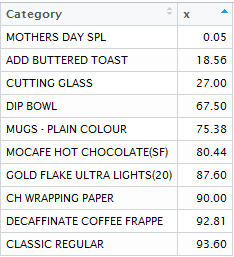


## Bottom 10 Item sales by Revenue

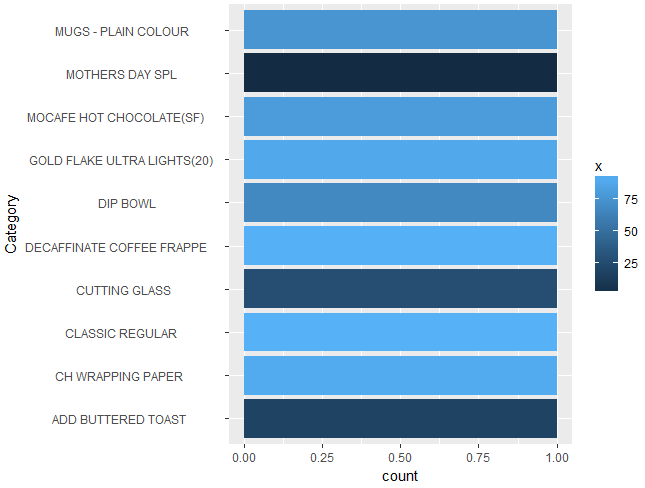
> temp3<-CafeGreatSalesAggSales %>%

+ top\_n(-10)

> View(temp3)



> ggplot(temp3, aes(Category,fill=x)) + geom\_bar()+coord\_flip()



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  | | --- | | **## TIME SERIES (Hourly and Monthly Analysis)**  > library(lubridate)> View(CafeGreatNewGroup)  > CafeGreatNewGroupTS1 <- ts(CafeGreatNewGroup[-1], start = as.Date("2010-04-01"), freq = 15)  > plot(CafeGreatNewGroupTS1) | |  | | |  | | --- | | > View(CafeGreatMonthGrpA)  > CafeGreatMonthGrpATS <- ts(CafeGreatMonthGrpA[-1], start = as.Date("2010-04-01"), freq = 12)  > plot(CafeGreatMonthGrpATS) | | |
|  |
| |  | | --- | |  | |

**## Market Basket Analysis Considering only FOOD Category**

> CafeGreatFood <- subset(CafeGreat, Category == "FOOD")

> View(CafeGreatFood)

> CafeGreatFood1<-CafeGreatFood[c(2,3,10)]

> View(CafeGreatFood1)

> library(splitstackshape)

> CafeGreatFood2 <- getanID(data = CafeGreatFood1, id.vars = c("Bill.Number","Category"))

> View(CafeGreatFood2)

> CafeGreatFood3<-CafeGreatFood2

> CafeGreatFood3$NewCategory <- paste(CafeGreatFood3$Category,CafeGreatFood3$.id)

> CafeGreatFoodData <- data.frame(CafeGreatFood3$Bill.Number,CafeGreatFood3$Item.Desc,CafeGreatFood3$NewCategory)

> timevar <- 'CafeGreatFood3.NewCategory'

> idvar <- 'CafeGreatFood3.Bill.Number'

> timevar

[1] "CafeGreatFood3.NewCategory"

> idvar

[1] "CafeGreatFood3.Bill.Number"

> wide <- reshape(CafeGreatFoodData,dir='w',idvar=idvar,timevar=timevar);

> attr(wide,'reshapeWide') <- NULL;

> View(wide)

> dim(wide)

[1] 32422 17

> CafeGreatFoodData1 <- data.frame(wide[c(-1)])

> View(CafeGreatFoodData1)

> rules <- apriori(CafeGreatFoodData1, parameter = list(supp = 0.0001,conf = 0.9))

Apriori

Parameter specification:

confidence minval smax arem aval originalSupport maxtime support minlen maxlen

0.9 0.1 1 none FALSE TRUE 5 1e-04 1 10

target ext

rules FALSE

Algorithmic control:

filter tree heap memopt load sort verbose

0.1 TRUE TRUE FALSE TRUE 2 TRUE

Absolute minimum support count: 3

set item appearances ...[0 item(s)] done [0.00s].

set transactions ...[1186 item(s), 32422 transaction(s)] done [0.01s].

sorting and recoding items ... [714 item(s)] done [0.00s].

creating transaction tree ... done [0.01s].

checking subsets of size 1 2 3 done [0.00s].

writing ... [7 rule(s)] done [0.00s].

creating S4 object ... done [0.00s].

> library(arules)

> itemFrequencyPlot(items(rules), support = 0.0001, cex.names = 0.9)



> options(digits=2)

> inspect(rules[1:7])

lhs rhs support confidence lift count

[1] {CafeGreatFood3.Item.Desc.FOOD.2=CHEDDAR CHEESE } => {CafeGreatFood3.Item.Desc.FOOD.1=MUSHROOM } 0.00031 1.00 1013 10

[2] {CafeGreatFood3.Item.Desc.FOOD.2=POTATO WEDGES } => {CafeGreatFood3.Item.Desc.FOOD.1=FRENCH FRIES } 0.00049 0.94 120 16

[3] {CafeGreatFood3.Item.Desc.FOOD.2=POTATO WEDGES ,

CafeGreatFood3.Item.Desc.FOOD.3=B.M.T. PANINI } => {CafeGreatFood3.Item.Desc.FOOD.1=FRENCH FRIES } 0.00012 1.00 127 4

[4] {CafeGreatFood3.Item.Desc.FOOD.1=SATAY CHICKEN PANINI ,

CafeGreatFood3.Item.Desc.FOOD.3=CHICKEN SALAMI PANINI } => {CafeGreatFood3.Item.Desc.FOOD.2=COUNTRY ROAST CHICKEN PANINI } 0.00015 1.00 81 5

[5] {CafeGreatFood3.Item.Desc.FOOD.1=B.M.T. PANINI ,

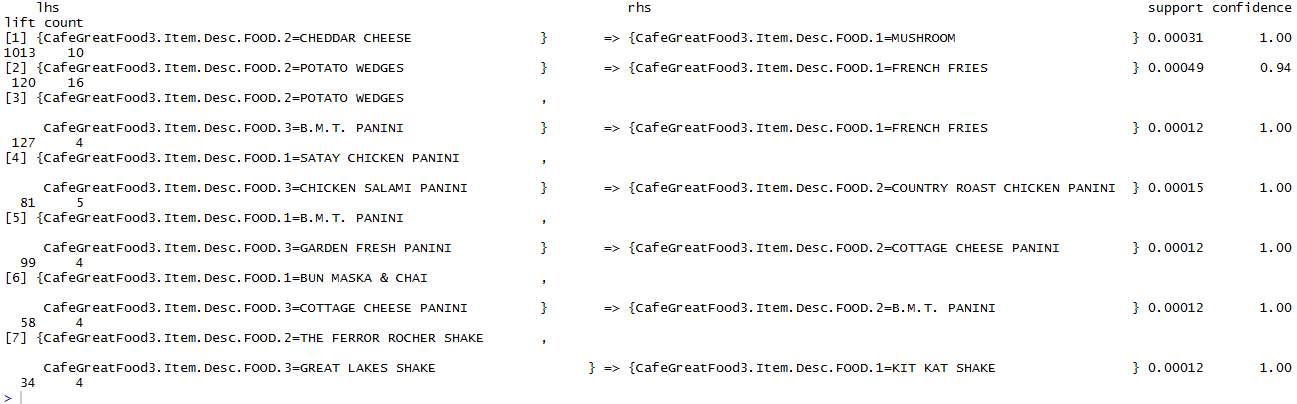
CafeGreatFood3.Item.Desc.FOOD.3=GARDEN FRESH PANINI } => {CafeGreatFood3.Item.Desc.FOOD.2=COTTAGE CHEESE PANINI } 0.00012 1.00 99 4

[6] {CafeGreatFood3.Item.Desc.FOOD.1=BUN MASKA & CHAI ,

CafeGreatFood3.Item.Desc.FOOD.3=COTTAGE CHEESE PANINI } => {CafeGreatFood3.Item.Desc.FOOD.2=B.M.T. PANINI } 0.00012 1.00 58 4

[7] {CafeGreatFood3.Item.Desc.FOOD.2=THE FERROR ROCHER SHAKE ,

CafeGreatFood3.Item.Desc.FOOD.3=GREAT LAKES SHAKE } => {CafeGreatFood3.Item.Desc.FOOD.1=KIT KAT SHAKE } 0.00012 1.00 34 4



> rules <- apriori(CafeGreatFoodData1, parameter = list(supp = 0.001, conf = 0.4))

Apriori

Parameter specification:

confidence minval smax arem aval originalSupport maxtime support minlen maxlen target ext

0.4 0.1 1 none FALSE TRUE 5 0.001 1 10 rules FALSE

Algorithmic control:

filter tree heap memopt load sort verbose

0.1 TRUE TRUE FALSE TRUE 2 TRUE

Absolute minimum support count: 32

set item appearances ...[0 item(s)] done [0.00s].

set transactions ...[1186 item(s), 32422 transaction(s)] done [0.00s].

sorting and recoding items ... [258 item(s)] done [0.00s].

creating transaction tree ... done [0.01s].

checking subsets of size 1 2 3 done [0.00s].

writing ... [1 rule(s)] done [0.00s].

creating S4 object ... done [0.00s].

> inspect(rules[1:1])

lhs rhs support confidence lift count

[1] {CafeGreatFood3.Item.Desc.FOOD.2=VANILLA ICECREAM } => {CafeGreatFood3.Item.Desc.FOOD.1=GREAT LAKES SHAKE } 0.001 0.47 4.4 34

**## Market Basket Analysis Considering All Category**

CafeGreatAll<-CafeGreat[c(2,3,10)]

> CafeGreatAll1 <- getanID(data = CafeGreatAll, id.vars = c("Bill.Number","Category"))

> CafeGreatAll2 <- CafeGreatAll1

> CafeGreatAll2$NewCategory <- paste(CafeGreatAll2$Category,CafeGreatAll2$.id)

> View(CafeGreatAll2)

> CafeGreatAllData <- data.frame(CafeGreatAll2$Bill.Number,CafeGreatAll2$Item.Desc,CafeGreatAll2$NewCategory)

> timevar <- 'CafeGreatAll2.NewCategory'

> idvar <- 'CafeGreatAll2.Bill.Number'

> wideAll <- reshape(CafeGreatAllData,dir='w',idvar=idvar,timevar=timevar);

> attr(wideAll,'reshapeWide') <- NULL

> View(wideAll)

> CafeGreatAllData1 <- data.frame(wideAll[c(-1)])

> rules1 <- apriori(CafeGreatAllData1, parameter = list(supp = 0.001, conf = 0.5))

Apriori

Parameter specification:

confidence minval smax arem aval originalSupport maxtime support minlen maxlen target ext

0.5 0.1 1 none FALSE TRUE 5 0.001 1 10 rules FALSE

Algorithmic control:

filter tree heap memopt load sort verbose

0.1 TRUE TRUE FALSE TRUE 2 TRUE

Absolute minimum support count: 69

set item appearances ...[0 item(s)] done [0.00s].

set transactions ...[2189 item(s), 69982 transaction(s)] done [0.02s].

sorting and recoding items ... [293 item(s)] done [0.00s].

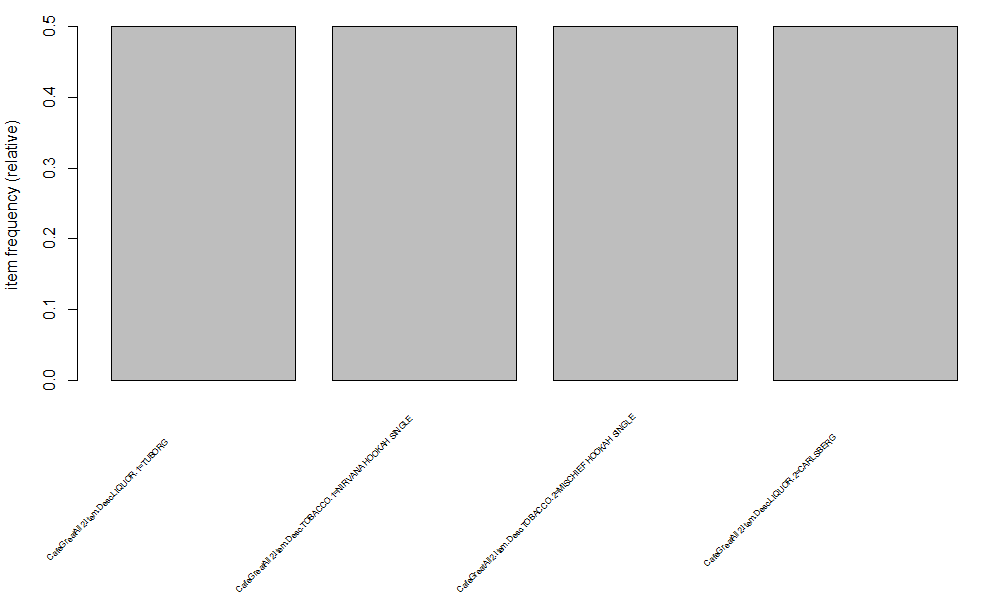
creating transaction tree ... done [0.02s].

checking subsets of size 1 2 3 done [0.00s].

writing ... [2 rule(s)] done [0.00s].

creating S4 object ... done [0.00s].

> itemFrequencyPlot(items(rules1), support = 0.0001, cex.names = 0.9)

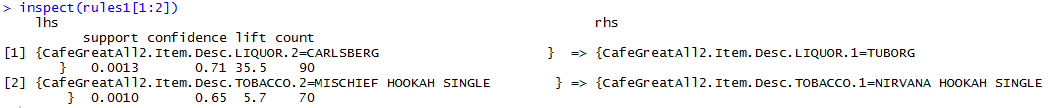


> inspect(rules1[1:2])

lhs rhs support confidence lift count

[1] {CafeGreatAll2.Item.Desc.LIQUOR.2=CARLSBERG } => {CafeGreatAll2.Item.Desc.LIQUOR.1=TUBORG } 0.0013 0.71 35.5 90

[2] {CafeGreatAll2.Item.Desc.TOBACCO.2=MISCHIEF HOOKAH SINGLE } => {CafeGreatAll2.Item.Desc.TOBACCO.1=NIRVANA HOOKAH SINGLE } 0.0010 0.65 5.7 70



> rules1 <- apriori(CafeGreatAllData1, parameter = list(supp = 0.0001, conf = 0.9))

Apriori

Parameter specification:

confidence minval smax arem aval originalSupport maxtime support minlen maxlen target ext

0.9 0.1 1 none FALSE TRUE 5 1e-04 1 10 rules FALSE

Algorithmic control:

filter tree heap memopt load sort verbose

0.1 TRUE TRUE FALSE TRUE 2 TRUE

Absolute minimum support count: 6

set item appearances ...[0 item(s)] done [0.00s].

set transactions ...[2189 item(s), 69982 transaction(s)] done [0.02s].

sorting and recoding items ... [1036 item(s)] done [0.00s].

creating transaction tree ... done [0.02s].

checking subsets of size 1 2 3 4 done [0.01s].

writing ... [8 rule(s)] done [0.00s].

creating S4 object ... done [0.00s].

> inspect(rules1[1:8])

lhs rhs support confidence lift count

[1] {CafeGreatAll2.Item.Desc.BEVERAGE.2=N R G HOOKAH } => {CafeGreatAll2.Item.Desc.BEVERAGE.1=2 RED BULL } 0.00019 1.00 3683.3 13

[2] {CafeGreatAll2.Item.Desc.FOOD.2=CHEDDAR CHEESE } => {CafeGreatAll2.Item.Desc.FOOD.1=MUSHROOM } 0.00014 1.00 2186.9 10

[3] {CafeGreatAll2.Item.Desc.LIQUOR...TPBACCO.2=BEER HOOKAH } => {CafeGreatAll2.Item.Desc.LIQUOR...TPBACCO.1=BEER HOOKAH } 0.00010 1.00 2186.9 7

[4] {CafeGreatAll2.Item.Desc.FOOD.2=POTATO WEDGES } => {CafeGreatAll2.Item.Desc.FOOD.1=FRENCH FRIES } 0.00023 0.94 258.3 16

[5] {CafeGreatAll2.Item.Desc.MISC.2=ADD ON S } => {CafeGreatAll2.Item.Desc.MISC.1=ADD ON S } 0.00036 0.93 184.6 25

[6] {CafeGreatAll2.Item.Desc.BEVERAGE.1=MOROCCAN MINT TEA ,

CafeGreatAll2.Item.Desc.BEVERAGE.2=QUA MINERAL WATER(500ML) } => {CafeGreatAll2.Item.Desc.TOBACCO.1=NIRVANA HOOKAH SINGLE } 0.00019 1.00 8.8 13

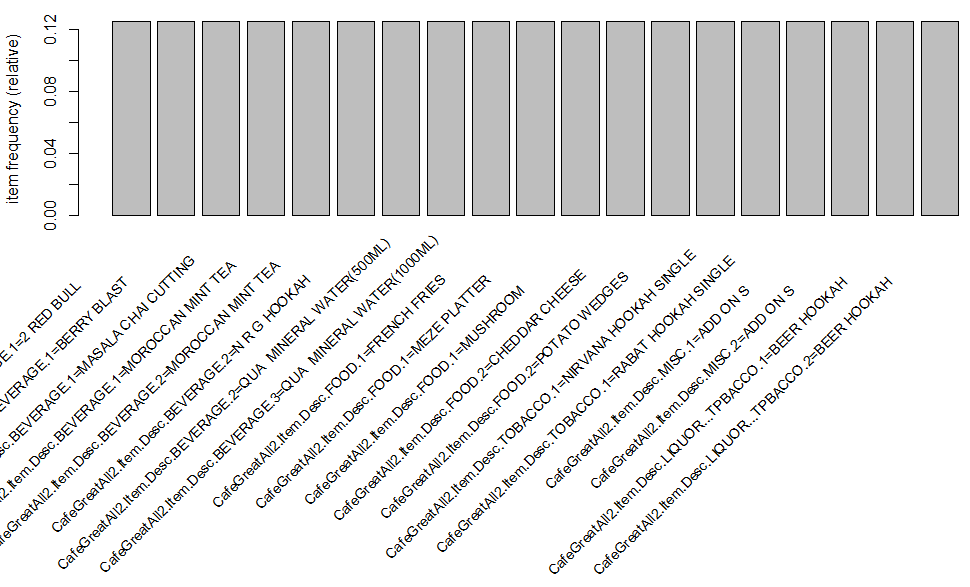
[7] {CafeGreatAll2.Item.Desc.BEVERAGE.1=MASALA CHAI CUTTING ,

CafeGreatAll2.Item.Desc.BEVERAGE.3=QUA MINERAL WATER(1000ML) } => {CafeGreatAll2.Item.Desc.BEVERAGE.2=MOROCCAN MINT TEA } 0.00010 1.00 244.7 7

[8] {CafeGreatAll2.Item.Desc.FOOD.1=MEZE PLATTER ,

CafeGreatAll2.Item.Desc.TOBACCO.1=RABAT HOOKAH SINGLE } => {CafeGreatAll2.Item.Desc.BEVERAGE.1=BERRY BLAST } 0.00016 1.00 56.6 11

> itemFrequencyPlot(items(rules1), support = 0.0001, cex.names = 0.9)



**### Identifying Price Change and Analysing**

CGTemp1 <- CafeGreat[c(1,3,5,6,9)]

View(CGTemp1)

CGTemp1$Item.Desc <- trimws(CGTemp1$Item.Desc,"r")

CGTemp1$Item.Desc <- trimws(CGTemp1$Item.Desc,"l")

CGTemp1$DateF <- as.POSIXct(CGTemp1$Date,format="%d-%b-%y")

CGTemp1$Mth <- month(CGTemp1$DateF)

CGTemp1Sorted <- CGTemp1[order(CGTemp1$Item.Desc,CGTemp1$DateF),]

View(CGTemp1Sorted)

CGTemp1SortedAgg <- aggregate(CGTemp1$Total, by=list(Category=CGTemp1$Item.Desc,CGTemp1$Mth,CGTemp1$Rate), FUN=sum)

CGTemp1SortedAggSrt <- CGTemp1SortedAgg[order(CGTemp1SortedAgg$Category,CGTemp1SortedAgg$Group.2,CGTemp1SortedAgg$Group.3),]

CGTemp1SortedAggSrt$Flg1 <- ifelse(CGTemp1SortedAggSrt$Category == lag(CGTemp1SortedAggSrt$Category), ifelse(CGTemp1SortedAggSrt$Group.3 != lag(CGTemp1SortedAggSrt$Group.3),ifelse(CGTemp1SortedAggSrt$Group.2 != lag(CGTemp1SortedAggSrt$Group.2),1,0),0),1)

CGTemp1SortedAggSrt1 <- CGTemp1SortedAggSrt

test1<-CGTemp1SortedAggSrt1[which((CGTemp1SortedAggSrt1$Category == lag(CGTemp1SortedAggSrt1$Category)) & (CGTemp1SortedAggSrt1$Flg1 == 1)),]

test2<-CGTemp1SortedAggSrt1[which((CGTemp1SortedAggSrt1$Category == lead(CGTemp1SortedAggSrt1$Category)) & (lead(CGTemp1SortedAggSrt1$Flg1) == 1)),]

new<-rbind(test1,test2)

newsort <- new[order(new$Category,new$Group.2,new$Group.3),]

newsort1 <- newsort

newsort1$Group.2 <- ifelse(newsort1$Category == lag(newsort1$Category),ifelse(newsort1$Group.3 == lag(newsort1$Group.3),lag(newsort1$Group.2),newsort1$Group.2),newsort1$Group.2)

newsort1$Group.2 <- ifelse(is.na(newsort1$Group.2),1,newsort1$Group.2)

newsort1Agg <- aggregate(newsort1$x, by=list(Category=newsort1$Category,newsort1$Group.2,newsort1$Group.3), FUN=sum)

newsort1Aggsort <- newsort1Agg[order(newsort1Agg$Category,newsort1Agg$Group.2,newsort1Agg$Group.3),]

View(newsort1Aggsort)

library(dplyr)

new.df <- newsort1Aggsort %>% group\_by(Category) %>% mutate(Index=1:n())

**## Assigning proper Column names**

colnames(new.df)[1] <- "Item"

colnames(new.df)[2] <- "Month"

colnames(new.df)[3] <- "Rate"

colnames(new.df)[4] <- "Sales"

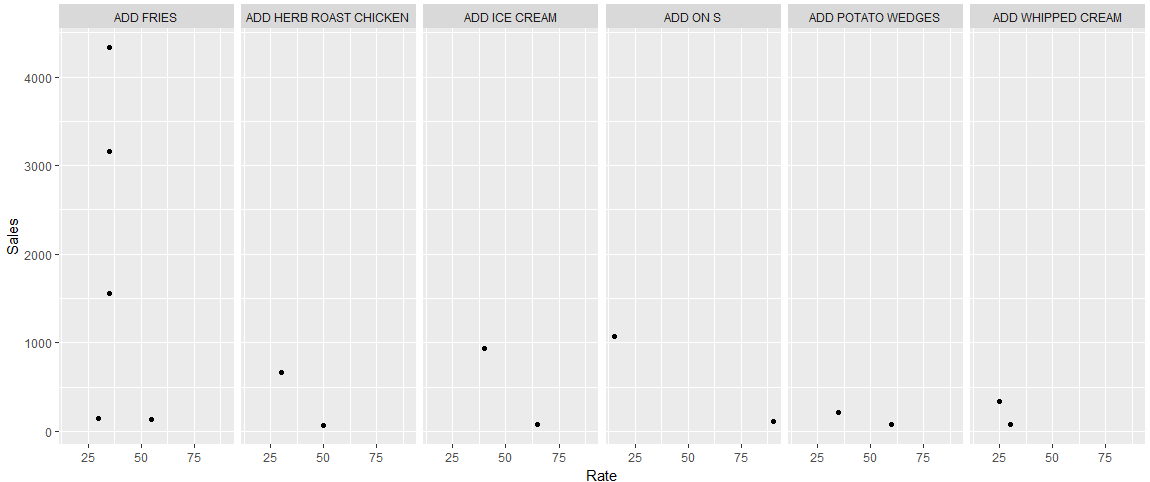
new.df1 <- new.df[1:21,]

new.df2 <- new.df1[1:16,]

**## Price Analysis Plot (Volume of Sales increases when Rate decreases)**

g <- ggplot(new.df2, aes(Rate, Sales))

g + geom\_point() + facet\_grid(. ~ Item)



**## Price Analysis Plot (Volume of Sales decreases when Rate decreases after certain amount for Al Sikandari Hookah Single )**

new.df3 <- new.df1[17:21,]

g <- ggplot(new.df3, aes(Rate, Sales))

g + geom\_point() + facet\_grid(. ~ Item)

