Untitled

data <- read\_xlsx("allnomods.xlsx")  
nrow(data)

## [1] 70883

str(data)

## Classes 'tbl\_df', 'tbl' and 'data.frame': 70883 obs. of 10 variables:  
## $ Date : num 20191223 20191109 20200121 20191124 20191127 ...  
## $ Transaction: num 34373 26286 39378 28987 29444 ...  
## $ Category : chr "Bev" "Bev" "Bev" "Bev" ...  
## $ Class : chr "Whiskey" "Sake" "Sake" "Promo Drinks" ...  
## $ ID : chr "20724" "20958" "20958" "21202" ...  
## $ Menu : chr "Yamazaki Btl" "Kubota Manju" "Kubota Manju" "Hato Masamune" ...  
## $ Qty : num 1 1 1 1 1 1 1 2 1 1 ...  
## $ Price : num 290 125 125 95 95 95 95 95 95 95 ...  
## $ Cost : num 0.466 0.576 0.576 0.468 0.621 ...  
## $ Row no. : num 14 2 7 16 4 13 7 3 9 8 ...

summary(data)

## Date Transaction Category Class   
## Min. :20191101 Min. :24796 Length:70883 Length:70883   
## 1st Qu.:20191123 1st Qu.:28785 Class :character Class :character   
## Median :20191215 Median :32721 Mode :character Mode :character   
## Mean :20193743 Mean :32665   
## 3rd Qu.:20200104 3rd Qu.:36527   
## Max. :20200130 Max. :40434   
## ID Menu Qty Price   
## Length:70883 Length:70883 Min. : 0.000 Min. : 0.00   
## Class :character Class :character 1st Qu.: 1.000 1st Qu.: 8.50   
## Mode :character Mode :character Median : 1.000 Median : 19.80   
## Mean : 1.067 Mean : 18.89   
## 3rd Qu.: 1.000 3rd Qu.: 27.80   
## Max. :21.000 Max. :290.00   
## Cost Row no.   
## Min. :0.0000 Min. : 1.000   
## 1st Qu.:0.2079 1st Qu.: 3.000   
## Median :0.2683 Median : 5.000   
## Mean :0.2427 Mean : 7.047   
## 3rd Qu.:0.3089 3rd Qu.: 9.000   
## Max. :0.6211 Max. :68.000

data$Date <- ymd(data$Date)  
data$Day <- wday(data$Date, label = T)  
head(data)

## # A tibble: 6 x 11  
## Date Transaction Category Class ID Menu Qty Price Cost `Row no.`  
## <date> <dbl> <chr> <chr> <chr> <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 2019-12-23 34373 Bev Whis~ 20724 Yama~ 1 290 0.466 14  
## 2 2019-11-09 26286 Bev Sake 20958 Kubo~ 1 125 0.576 2  
## 3 2020-01-21 39378 Bev Sake 20958 Kubo~ 1 125 0.576 7  
## 4 2019-11-24 28987 Bev Prom~ 21202 Hato~ 1 95 0.468 16  
## 5 2019-11-27 29444 Bev Prom~ 21201 Rokk~ 1 95 0.621 4  
## 6 2019-12-05 30950 Bev Prom~ 21201 Rokk~ 1 95 0.621 13  
## # ... with 1 more variable: Day <ord>

new\_vars <- c("Date","Transaction","Category", "Class", "ID","Menu","Qty","Price","Cost","Row","Day")  
colnames(data) <- new\_vars  
summary(data)

## Date Transaction Category Class   
## Min. :2019-11-01 Min. :24796 Length:70883 Length:70883   
## 1st Qu.:2019-11-23 1st Qu.:28785 Class :character Class :character   
## Median :2019-12-15 Median :32721 Mode :character Mode :character   
## Mean :2019-12-14 Mean :32665   
## 3rd Qu.:2020-01-04 3rd Qu.:36527   
## Max. :2020-01-30 Max. :40434   
##   
## ID Menu Qty Price   
## Length:70883 Length:70883 Min. : 0.000 Min. : 0.00   
## Class :character Class :character 1st Qu.: 1.000 1st Qu.: 8.50   
## Mode :character Mode :character Median : 1.000 Median : 19.80   
## Mean : 1.067 Mean : 18.89   
## 3rd Qu.: 1.000 3rd Qu.: 27.80   
## Max. :21.000 Max. :290.00   
##   
## Cost Row Day   
## Min. :0.0000 Min. : 1.000 Sun:11991   
## 1st Qu.:0.2079 1st Qu.: 3.000 Mon: 8655   
## Median :0.2683 Median : 5.000 Tue: 8803   
## Mean :0.2427 Mean : 7.047 Wed: 9202   
## 3rd Qu.:0.3089 3rd Qu.: 9.000 Thu: 9353   
## Max. :0.6211 Max. :68.000 Fri:10490   
## Sat:12389

# arrange columns  
colnames(data)

## [1] "Date" "Transaction" "Category" "Class" "ID"   
## [6] "Menu" "Qty" "Price" "Cost" "Row"   
## [11] "Day"

data <- data[, c(1,11,2,3,4,5,6,7,8,9,10)]  
summary(data)

## Date Day Transaction Category   
## Min. :2019-11-01 Sun:11991 Min. :24796 Length:70883   
## 1st Qu.:2019-11-23 Mon: 8655 1st Qu.:28785 Class :character   
## Median :2019-12-15 Tue: 8803 Median :32721 Mode :character   
## Mean :2019-12-14 Wed: 9202 Mean :32665   
## 3rd Qu.:2020-01-04 Thu: 9353 3rd Qu.:36527   
## Max. :2020-01-30 Fri:10490 Max. :40434   
## Sat:12389   
## Class ID Menu Qty   
## Length:70883 Length:70883 Length:70883 Min. : 0.000   
## Class :character Class :character Class :character 1st Qu.: 1.000   
## Mode :character Mode :character Mode :character Median : 1.000   
## Mean : 1.067   
## 3rd Qu.: 1.000   
## Max. :21.000   
##   
## Price Cost Row   
## Min. : 0.00 Min. :0.0000 Min. : 1.000   
## 1st Qu.: 8.50 1st Qu.:0.2079 1st Qu.: 3.000   
## Median : 19.80 Median :0.2683 Median : 5.000   
## Mean : 18.89 Mean :0.2427 Mean : 7.047   
## 3rd Qu.: 27.80 3rd Qu.:0.3089 3rd Qu.: 9.000   
## Max. :290.00 Max. :0.6211 Max. :68.000   
##

# Set factors  
cols <- c("Date","Transaction","Category", "Class","Menu")  
data <- data %>% mutate\_at(cols, factor)  
head(data)

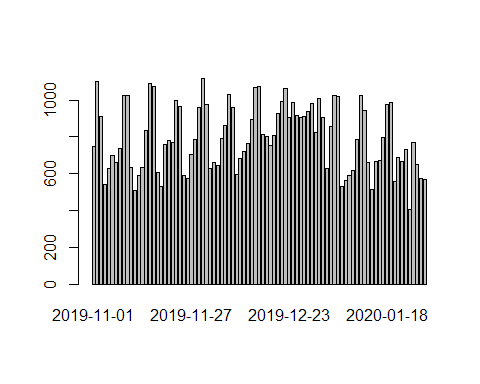
## # A tibble: 6 x 11  
## Date Day Transaction Category Class ID Menu Qty Price Cost Row  
## <fct> <ord> <fct> <fct> <fct> <chr> <fct> <dbl> <dbl> <dbl> <dbl>  
## 1 2019-~ Mon 34373 Bev Whiskey 20724 Yamaz~ 1 290 0.466 14  
## 2 2019-~ Sat 26286 Bev Sake 20958 Kubot~ 1 125 0.576 2  
## 3 2020-~ Tue 39378 Bev Sake 20958 Kubot~ 1 125 0.576 7  
## 4 2019-~ Sun 28987 Bev Promo ~ 21202 Hato ~ 1 95 0.468 16  
## 5 2019-~ Wed 29444 Bev Promo ~ 21201 Rokko~ 1 95 0.621 4  
## 6 2019-~ Thu 30950 Bev Promo ~ 21201 Rokko~ 1 95 0.621 13

#filter data for rules  
data <- as\_tibble(data)  
transdata <- select(data, Transaction, Menu)

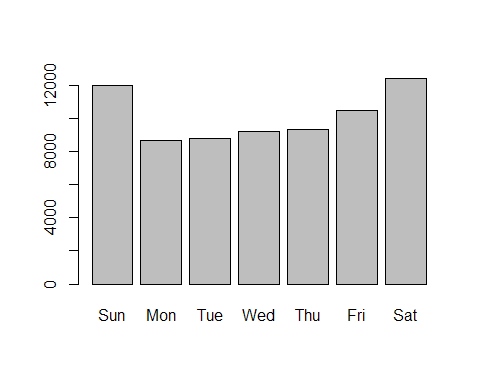
quantile(data$Price,c(0.01,0.02,0.03,0.1,0.2,0.3,0.4,0.50,0.6,0.7,0.8,0.9,0.95,0.99,1)) # significant difference in 0% and 1%

## 1% 2% 3% 10% 20% 30% 40% 50% 60% 70% 80%   
## 0.00 0.00 0.00 3.00 6.80 9.80 13.80 19.80 23.80 26.80 29.74   
## 90% 95% 99% 100%   
## 32.80 36.80 45.80 290.00

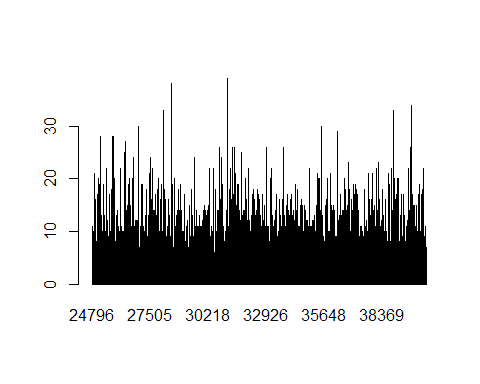
attach(data)  
plot(Date)



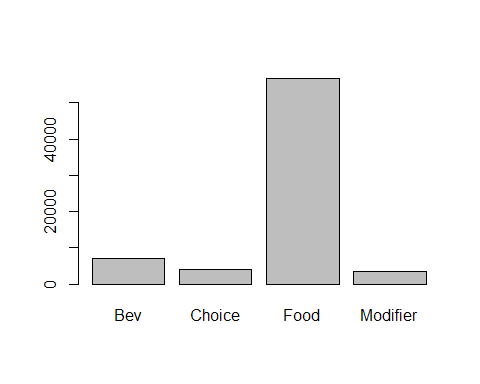
plot(Day)



plot(Transaction)

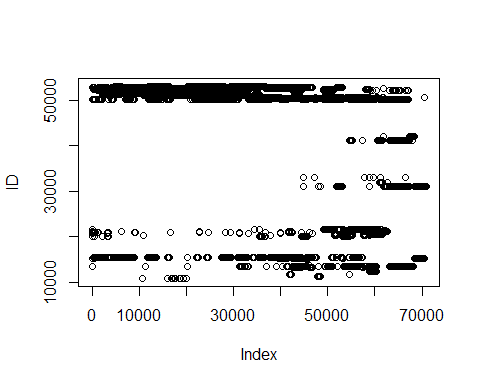


plot(Category)

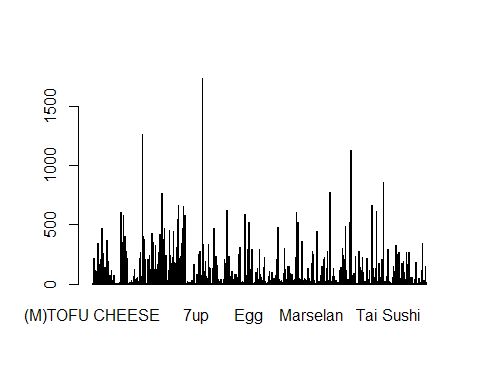


plot(ID)

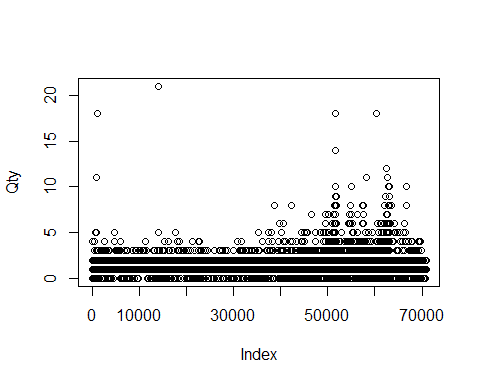
## Warning in xy.coords(x, y, xlabel, ylabel, log): NAs introduced by coercion



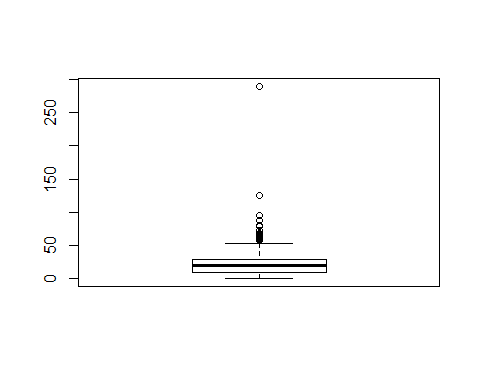
plot(Menu)



plot(Qty)

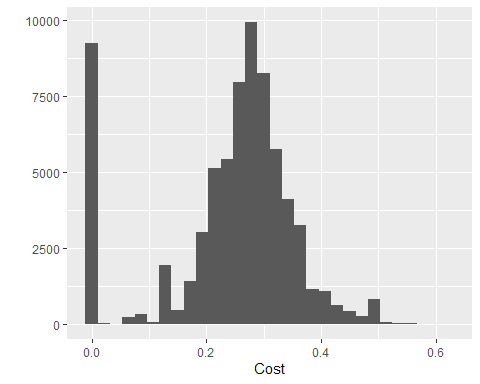


boxplot(Price)

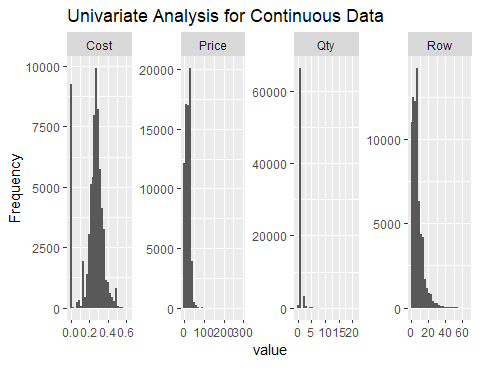


qplot(Cost)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

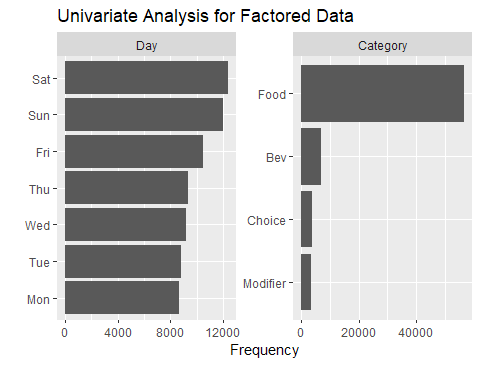


plot\_histogram(data, title ="Univariate Analysis for Continuous Data")

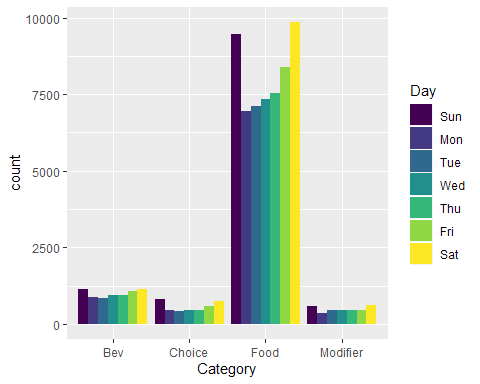


plot\_bar(data, title ="Univariate Analysis for Factored Data")

## 5 columns ignored with more than 50 categories.  
## Date: 89 categories  
## Transaction: 15562 categories  
## Class: 53 categories  
## ID: 708 categories  
## Menu: 702 categories



ggplot(data = data) +   
 geom\_bar(mapping = aes(x = Category, fill = Day), position = "dodge")



Agg.RTxn <- split(transdata$Menu ,transdata$Transaction)  
class(Agg.RTxn)

## [1] "list"

## To see specific row number transaction  
Agg.RTxn [35]

## $`24830`  
## [1] [SL]Buta Don Set muroka nama summer  
## 702 Levels: (M)TOFU CHEESE (S)TOFU CHEESE ... Zuwaigani Sushi

## logic to remove duplicate items from the list  
Agg.RTxn\_DD <- list()  
for (i in 1:length(Agg.RTxn)) {  
 Agg.RTxn\_DD[[i]] <- as.character(Agg.RTxn[[i]][!duplicated(Agg.RTxn[[i]])])  
}  
## converting transaction items from list format to transaction format  
Txns <- as(Agg.RTxn\_DD, "transactions")  
  
summary(Txns)

## transactions as itemMatrix in sparse format with  
## 15562 rows (elements/itemsets/transactions) and  
## 702 columns (items) and a density of 0.006221232   
##   
## most frequent items:  
## Add KM Soup [S] Mini Chawan Rice SUN Tofu Cheesecake   
## 1262 984 975 857   
## Nokke Don (Other)   
## 652 63234   
##   
## element (itemset/transaction) length distribution:  
## sizes  
## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16   
## 1387 2911 3095 2517 1779 1215 818 564 342 256 196 129 92 68 49 37   
## 17 18 19 20 21 22 23 24 25 26 28 30 31 36   
## 27 21 17 6 9 6 4 4 3 3 2 1 2 2   
##   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 2.000 4.000 4.367 5.000 36.000   
##   
## includes extended item information - examples:  
## labels  
## 1 (M)TOFU CHEESE  
## 2 (S)TOFU CHEESE  
## 3 [2p]Salmon Sashimi 3K

inspect(Txns[1:5])

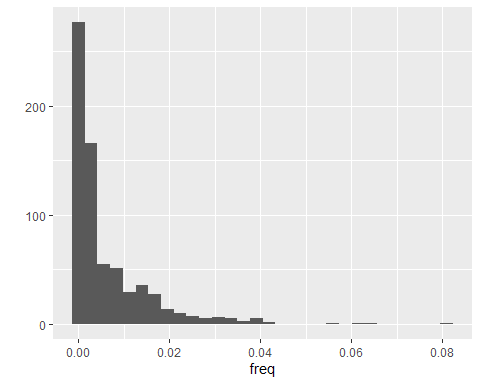
## items   
## [1] {[SL]Beef Nabe Set,   
## Jyako Katsu}   
## [2] {[4pc]Avo Soft Shell,   
## Passionfruit Highball,   
## Tokushima Ramen}   
## [3] {[SL]Gindara & Niku Tofu Nabe,  
## Teriyaki}   
## [4] {[SL]US Tenderloin Steak,   
## Miso Soup,   
## Shokado Bento}   
## [5] {[S] Mini Chawan,   
## [SL]Saba & Salmon Teriyaki,   
## [SL]Unagi Yakiniku KM,   
## Add KM Soup,   
## Hamachi Kama Yaki}

freq <- itemFrequency(Txns)  
freq <- freq[order(-freq)]  
freq["Matcha Parfait"]

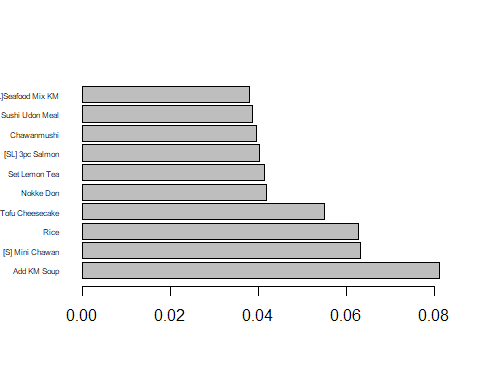
## Matcha Parfait   
## 0.01773551

qplot(freq)

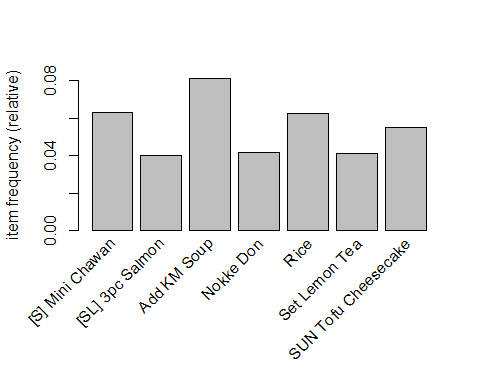
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



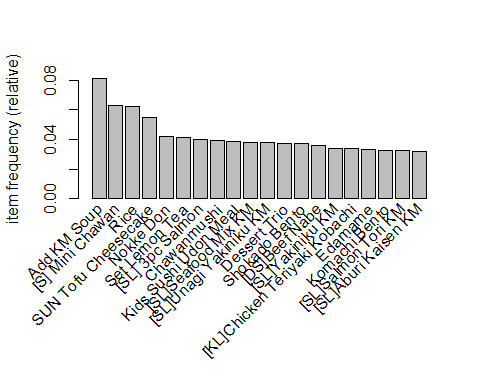
barplot(freq[1:10], horiz = TRUE, axisnames = TRUE, las=1, cex.names=0.5)



itemFrequencyPlot(Txns, support = 0.04)



itemFrequencyPlot(Txns, topN = 20)



library("arulesViz")  
  
arules1 <- apriori(data = Txns, parameter=list(supp= 0.001, conf = 0.6, minlen = 2))

## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## 0.6 0.1 1 none FALSE TRUE 5 0.001 2  
## maxlen target ext  
## 10 rules FALSE  
##   
## Algorithmic control:  
## filter tree heap memopt load sort verbose  
## 0.1 TRUE TRUE FALSE TRUE 2 TRUE  
##   
## Absolute minimum support count: 15   
##   
## set item appearances ...[0 item(s)] done [0.00s].  
## set transactions ...[702 item(s), 15562 transaction(s)] done [0.01s].  
## sorting and recoding items ... [455 item(s)] done [0.00s].  
## creating transaction tree ... done [0.00s].  
## checking subsets of size 1 2 3 4 done [0.00s].  
## writing ... [93 rule(s)] done [0.00s].  
## creating S4 object ... done [0.00s].

summary(arules1)

## set of 93 rules  
##   
## rule length distribution (lhs + rhs):sizes  
## 2 3 4   
## 18 68 7   
##   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 2.000 3.000 3.000 2.882 3.000 4.000   
##   
## summary of quality measures:  
## support confidence lift count   
## Min. :0.001028 Min. :0.6000 Min. : 7.399 Min. : 16.00   
## 1st Qu.:0.001157 1st Qu.:0.6296 1st Qu.: 8.504 1st Qu.: 18.00   
## Median :0.001349 Median :0.6800 Median : 22.779 Median : 21.00   
## Mean :0.002165 Mean :0.7335 Mean : 33.844 Mean : 33.69   
## 3rd Qu.:0.002378 3rd Qu.:0.8077 3rd Qu.: 33.744 3rd Qu.: 37.00   
## Max. :0.015165 Max. :1.0000 Max. :420.595 Max. :236.00   
##   
## mining info:  
## data ntransactions support confidence  
## Txns 15562 0.001 0.6

inspect(sort(arules1,by="support"))

## lhs rhs support confidence lift count  
## [1] {Teriyaki} => {[SL]Gindara & Niku Tofu Nabe} 0.015165146 0.9516129 33.656818 236  
## [2] {Niku Atsu Hotate} => {Nokke Don} 0.008032387 1.0000000 23.868098 125  
## [3] {[SL]Premium Seafood KM} => {Add KM Soup} 0.006682946 0.6500000 8.015293 104  
## [4] {[SD]Aus Sirloin Steak} => {Red Wine} 0.006040355 0.7833333 25.826766 94  
## [5] {JP Wagyu Steak} => {Red Wine} 0.005783318 0.6521739 21.502395 90  
## [6] {JP Wagyu Steak} => {Black Pepper} 0.005526282 0.6231884 23.425261 86  
## [7] {[SD]US Tenderloin Set} => {Red Wine} 0.005269246 0.6507937 21.456887 82  
## [8] {[SD]Japanese Wagyu Set} => {Red Wine} 0.005012209 0.7155963 23.593454 78  
## [9] {[SD]US Tenderloin Set} => {Black Pepper} 0.005012209 0.6190476 23.269611 78  
## [10] {[SD]Japanese Wagyu Set} => {Black Pepper} 0.004755173 0.6788991 25.519390 74  
## [11] {[S] Mini Chawan,   
## Teriyaki} => {[SL]Gindara & Niku Tofu Nabe} 0.004176841 0.9848485 34.832300 65  
## [12] {[SD]Aus Sirloin Steak,   
## Black Pepper} => {Red Wine} 0.003148696 0.7101449 23.413719 49  
## [13] {[SD]Japanese Wagyu Set,   
## Black Pepper} => {Red Wine} 0.003020177 0.6351351 20.940621 47  
## [14] {[SD]Japanese Wagyu Set,   
## Red Wine} => {Black Pepper} 0.003020177 0.6025641 22.650006 47  
## [15] {[S] Mini Chawan,   
## [SL]Aburi Kaisen KM} => {Add KM Soup} 0.003020177 0.6351351 7.831991 47  
## [16] {[SL]Seafood Mix KM,   
## [SL]Unagi Yakiniku KM} => {Add KM Soup} 0.002827400 0.6027397 7.432516 44  
## [17] {[SL] 3pc Salmon,   
## Teriyaki} => {[SL]Gindara & Niku Tofu Nabe} 0.002698882 0.9545455 33.760537 42  
## [18] {[S] Mini Chawan,   
## [SL]Tori Unagi KM} => {Add KM Soup} 0.002698882 0.6176471 7.616342 42  
## [19] {[SL]Salmon Tori KM,   
## [SL]Unagi Yakiniku KM} => {Add KM Soup} 0.002570364 0.6060606 7.473467 40  
## [20] {[SD]Steak Moriawase} => {Red Wine} 0.002506105 0.6393443 21.079397 39  
## [21] {US Tenderloin Steak} => {Red Wine} 0.002506105 0.6610169 21.793953 39  
## [22] {Steak Moriawase} => {Red Wine} 0.002441846 0.6909091 22.779507 38  
## [23] {US Tenderloin Steak} => {Black Pepper} 0.002377586 0.6271186 23.572996 37  
## [24] {[SD]Aus Sirloin Steak,   
## Goma Mustard} => {Red Wine} 0.002377586 0.7254902 23.919658 37  
## [25] {[SL]Salmon Tori KM,   
## [SL]Yakiniku KM} => {Add KM Soup} 0.002377586 0.6727273 8.295548 37  
## [26] {Seasonal Sake 150ml} => {Seasonal Sake Appetizer} 0.002313327 1.0000000 420.594595 36  
## [27] {Seasonal Sake Appetizer} => {Seasonal Sake 150ml} 0.002313327 0.9729730 420.594595 36  
## [28] {[SL]Seafood Mix KM,   
## [SL]Tori Unagi KM} => {Add KM Soup} 0.002249068 0.7000000 8.631854 35  
## [29] {[SL]Unagi Yakiniku KM,   
## [SL]Yakiniku KM} => {Add KM Soup} 0.002249068 0.6250000 7.707013 35  
## [30] {Add KM Soup,   
## Teriyaki} => {[SL]Gindara & Niku Tofu Nabe} 0.002056291 0.9696970 34.296419 32  
## [31] {[SL]Aburi Kaisen KM,   
## [SL]Yakiniku KM} => {Add KM Soup} 0.001992032 0.6078431 7.495448 31  
## [32] {[SD]Aus Sirloin Steak,   
## Ponzu} => {Red Wine} 0.001863514 0.6304348 20.785648 29  
## [33] {Set Lemon Tea,   
## Teriyaki} => {[SL]Gindara & Niku Tofu Nabe} 0.001863514 0.9666667 34.189242 29  
## [34] {[SL]Aburi Kaisen KM,   
## [SL]Niku Atsu Hotate Unagi KM} => {Add KM Soup} 0.001863514 0.6041667 7.450112 29  
## [35] {Aust Sirloin Steak} => {Red Wine} 0.001734996 0.6923077 22.825619 27  
## [36] {[SD]Japanese Wagyu Set,   
## Goma Mustard} => {Red Wine} 0.001734996 0.6585366 21.712174 27  
## [37] {[SL]Unagi Yakiniku KM,   
## Set PassionfruitTea} => {Add KM Soup} 0.001670736 0.7027027 8.665182 26  
## [38] {[SL] 3pc Salmon,   
## [SL]Salmon Tori KM} => {Add KM Soup} 0.001670736 0.6190476 7.633613 26  
## [39] {Black Pepper,   
## SUN Tofu Cheesecake} => {Red Wine} 0.001477959 0.6388889 21.064383 23  
## [40] {Black Pepper,   
## Goma Mustard,   
## Ponzu} => {Red Wine} 0.001477959 0.9200000 30.332712 23  
## [41] {Goma Mustard,   
## Ponzu,   
## Red Wine} => {Black Pepper} 0.001477959 0.8518519 32.020576 23  
## [42] {Black Pepper,   
## Goma Mustard,   
## Red Wine} => {Ponzu} 0.001477959 0.6571429 36.523061 23  
## [43] {Black Pepper,   
## Ponzu,   
## Red Wine} => {Goma Mustard} 0.001477959 0.7666667 42.762963 23  
## [44] {[SD]Steak Moriawase,   
## Black Pepper} => {Red Wine} 0.001413700 0.6111111 20.148540 22  
## [45] {[SL]Seafood Mix KM,   
## Set PassionfruitTea} => {Add KM Soup} 0.001413700 0.6470588 7.979025 22  
## [46] {[SL]Aburi Kaisen KM,   
## [SL]Tori Unagi KM} => {Add KM Soup} 0.001349441 0.6176471 7.616342 21  
## [47] {[SL]Niku Atsu Hotate Unagi KM,   
## [SL]Salmon Tori KM} => {Add KM Soup} 0.001349441 0.8076923 9.959832 21  
## [48] {[SD]Sangenton Pork Steak Set} => {Black Pepper} 0.001285182 0.7407407 27.843979 20  
## [49] {[SL]Salmon Sushi Mori,   
## Sushi Rice} => {[2p]Sashimi 5K} 0.001285182 1.0000000 72.046296 20  
## [50] {[2p]Sashimi 5K,   
## Sushi Rice} => {[SL]Salmon Sushi Mori} 0.001285182 0.6451613 60.119760 20  
## [51] {[2p]Sashimi 5K,   
## [SL]Salmon Sushi Mori} => {Sushi Rice} 0.001285182 0.7407407 198.748404 20  
## [52] {[SL]Una Don,   
## Teriyaki} => {[SL]Gindara & Niku Tofu Nabe} 0.001285182 1.0000000 35.368182 20  
## [53] {[SL]Gindara & Niku Tofu Nabe,   
## [SL]Una Don} => {Teriyaki} 0.001285182 0.6896552 43.275862 20  
## [54] {[KL]Pork Shogayaki Kobachi,   
## [SL] 3pc Salmon} => {[S] Mini Chawan} 0.001285182 0.6666667 10.543360 20  
## [55] {[SL]Niku Atsu Hotate Unagi KM,   
## [SL]Yakiniku KM} => {Add KM Soup} 0.001285182 0.6896552 8.504290 20  
## [56] {[SL]Aburi Kaisen KM,   
## SUN Tofu Cheesecake} => {Add KM Soup} 0.001285182 0.7692308 9.485554 20  
## [57] {Miso Soup,   
## Sushi Rice} => {[2p]Sashimi 5K} 0.001220923 0.9500000 68.443981 19  
## [58] {[2p]Sashimi 5K,   
## Sushi Rice} => {Miso Soup} 0.001220923 0.6129032 48.663265 19  
## [59] {[SL]Nama Bara Chirashi Set,   
## [SL]Seafood Mix KM} => {Add KM Soup} 0.001220923 0.6551724 8.079075 19  
## [60] {[SD]Saba Shio & Pork Toji,   
## Black Pepper} => {Red Wine} 0.001220923 0.8260870 27.236367 19  
## [61] {[SD]Saba Shio & Pork Toji,   
## Red Wine} => {Black Pepper} 0.001220923 0.6785714 25.507074 19  
## [62] {[SL]Gindara & Niku Tofu Nabe,   
## [SL]Unagi Yakiniku KM} => {Add KM Soup} 0.001220923 0.6551724 8.079075 19  
## [63] {[KL]Chicken Teriyaki Kobachi,   
## [SL]Unagi Yakiniku KM} => {Add KM Soup} 0.001220923 0.6333333 7.809773 19  
## [64] {[SL]Yakiniku KM,   
## SUN Tofu Cheesecake} => {Add KM Soup} 0.001220923 0.6785714 8.367614 19  
## [65] {Black Pepper,   
## Surume Ika Teriyaki} => {Red Wine} 0.001220923 0.6333333 20.881215 19  
## [66] {[SL]Unagi Yakiniku KM,   
## SUN Tofu Cheesecake} => {Add KM Soup} 0.001220923 0.6129032 7.557845 19  
## [67] {Black Pepper,   
## Mixed Roll Sushi} => {Red Wine} 0.001220923 0.7037037 23.201350 19  
## [68] {Mixed Roll Sushi,   
## Red Wine} => {Black Pepper} 0.001220923 0.6129032 23.038647 19  
## [69] {Black Pepper,   
## Dessert Trio} => {Red Wine} 0.001220923 0.9047619 29.830307 19  
## [70] {[S] Mini Chawan,   
## [SL]Premium Seafood KM} => {Add KM Soup} 0.001156664 0.8181818 10.089180 18  
## [71] {[2p]Sashimi 7K,   
## Miso Soup} => {[2p]Sashimi 5K} 0.001156664 0.8181818 58.946970 18  
## [72] {[2p]Sashimi 5K,   
## [2p]Sashimi 7K} => {Miso Soup} 0.001156664 0.7826087 62.137533 18  
## [73] {Shokado Bento,   
## Teriyaki} => {[SL]Gindara & Niku Tofu Nabe} 0.001156664 0.9000000 31.831364 18  
## [74] {[SL] 3pc Salmon,   
## [SL]Teriyaki Wagyu Harami Don} => {[S] Mini Chawan} 0.001156664 0.6000000 9.489024 18  
## [75] {[SL]Seafood Mix KM,   
## SUN Tofu Cheesecake} => {Add KM Soup} 0.001156664 0.6000000 7.398732 18  
## [76] {[SL]Salmon Sushi Mori,   
## Miso Soup} => {[2p]Sashimi 5K} 0.001092405 0.9444444 68.043724 17  
## [77] {[2p]Sashimi 5K,   
## [SL]Salmon Sushi Mori} => {Miso Soup} 0.001092405 0.6296296 49.991308 17  
## [78] {[KL]Chicken Teriyaki Kobachi,   
## Teriyaki} => {[SL]Gindara & Niku Tofu Nabe} 0.001092405 1.0000000 35.368182 17  
## [79] {[KL]Chicken Teriyaki Kobachi,   
## [SL]Gindara & Niku Tofu Nabe} => {Teriyaki} 0.001092405 0.6800000 42.670000 17  
## [80] {[SL]Yakiniku KM,   
## Teriyaki} => {[SL]Gindara & Niku Tofu Nabe} 0.001092405 0.9444444 33.403283 17  
## [81] {[SL]Teriyaki Wagyu Harami Don,   
## [SL]Yakiniku KM} => {Add KM Soup} 0.001092405 0.6296296 7.764102 17  
## [82] {[SD]Premium Seafood KM,   
## Ponzu} => {Red Wine} 0.001092405 0.7391304 24.369381 17  
## [83] {[SD]Premium Seafood KM,   
## Red Wine} => {Ponzu} 0.001092405 0.6071429 33.744133 17  
## [84] {[KL]Pork Shogayaki Kobachi,   
## [SL]Yakiniku KM} => {Add KM Soup} 0.001092405 0.7391304 9.114380 17  
## [85] {[SL]Beef Nabe Set,   
## [SL]Yakiniku KM} => {Add KM Soup} 0.001092405 0.6800000 8.385230 17  
## [86] {Goma Mustard,   
## Rice} => {Red Wine} 0.001092405 0.6296296 20.759102 17  
## [87] {[S] Mini Chawan,   
## [SL] 3pc Salmon,   
## Teriyaki} => {[SL]Gindara & Niku Tofu Nabe} 0.001092405 1.0000000 35.368182 17  
## [88] {[S] Mini Chawan,   
## [SL] 3pc Salmon,   
## [SL]Yakiniku KM} => {Add KM Soup} 0.001092405 0.6800000 8.385230 17  
## [89] {[S] Mini Chawan,   
## [SL] 3pc Salmon,   
## [SL]Unagi Yakiniku KM} => {Add KM Soup} 0.001092405 0.6296296 7.764102 17  
## [90] {[SL]Unagi Yakiniku KM,   
## Teriyaki} => {[SL]Gindara & Niku Tofu Nabe} 0.001028145 1.0000000 35.368182 16  
## [91] {[SL]Aburi Kaisen KM,   
## Set PassionfruitTea} => {Add KM Soup} 0.001028145 0.7272727 8.968160 16  
## [92] {[SL]Yakiniku KM,   
## Komachi Bento} => {Add KM Soup} 0.001028145 0.7619048 9.395215 16  
## [93] {[KL]Chicken Teriyaki Kobachi,   
## [SL]Aburi Kaisen KM} => {Add KM Soup} 0.001028145 0.6153846 7.588443 16

arules2 <- apriori(data = Txns, parameter = list(support = 0.01, confidence = 0.3, minlen = 2))

## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## 0.3 0.1 1 none FALSE TRUE 5 0.01 2  
## maxlen target ext  
## 10 rules FALSE  
##   
## Algorithmic control:  
## filter tree heap memopt load sort verbose  
## 0.1 TRUE TRUE FALSE TRUE 2 TRUE  
##   
## Absolute minimum support count: 155   
##   
## set item appearances ...[0 item(s)] done [0.00s].  
## set transactions ...[702 item(s), 15562 transaction(s)] done [0.01s].  
## sorting and recoding items ... [153 item(s)] done [0.00s].  
## creating transaction tree ... done [0.00s].  
## checking subsets of size 1 2 3 done [0.00s].  
## writing ... [12 rule(s)] done [0.00s].  
## creating S4 object ... done [0.00s].

summary(arules2)

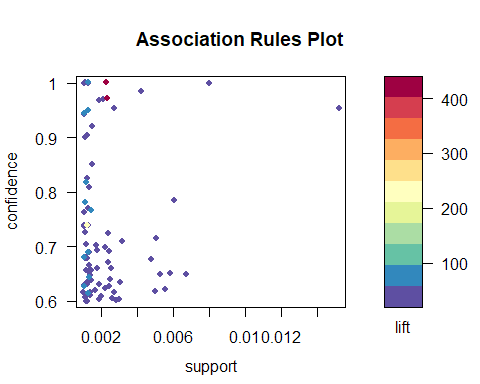
## set of 12 rules  
##   
## rule length distribution (lhs + rhs):sizes  
## 2   
## 12   
##   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 2 2 2 2 2 2   
##   
## summary of quality measures:  
## support confidence lift count   
## Min. :0.01176 Min. :0.3562 Min. : 5.613 Min. :183.0   
## 1st Qu.:0.01481 1st Qu.:0.4622 1st Qu.: 5.700 1st Qu.:230.5   
## Median :0.01520 Median :0.4975 Median : 6.135 Median :236.5   
## Mean :0.01526 Mean :0.5245 Mean :12.694 Mean :237.4   
## 3rd Qu.:0.01611 3rd Qu.:0.5277 3rd Qu.:18.874 3rd Qu.:250.8   
## Max. :0.01754 Max. :0.9516 Max. :33.657 Max. :273.0   
##   
## mining info:  
## data ntransactions support confidence  
## Txns 15562 0.01 0.3

inspect(sort(arules2,by="confidence"))

## lhs rhs   
## [1] {Teriyaki} => {[SL]Gindara & Niku Tofu Nabe}  
## [2] {Black Pepper} => {Red Wine}   
## [3] {[SL]Gindara & Niku Tofu Nabe} => {Teriyaki}   
## [4] {[SL]Niku Atsu Hotate Unagi KM} => {Add KM Soup}   
## [5] {Red Wine} => {Black Pepper}   
## [6] {[SL]Yakiniku KM} => {Add KM Soup}   
## [7] {[SL]Aburi Kaisen KM} => {Add KM Soup}   
## [8] {[SL]Tori Unagi KM} => {Add KM Soup}   
## [9] {[SL]Salmon Tori KM} => {Add KM Soup}   
## [10] {[SL]Unagi Yakiniku KM} => {Add KM Soup}   
## [11] {[SL]Seafood Mix KM} => {Add KM Soup}   
## [12] {[SL] 3pc Salmon} => {[S] Mini Chawan}   
## support confidence lift count  
## [1] 0.01516515 0.9516129 33.656818 236   
## [2] 0.01522940 0.5724638 18.874324 237   
## [3] 0.01516515 0.5363636 33.656818 236   
## [4] 0.01355867 0.5248756 6.472357 211   
## [5] 0.01522940 0.5021186 18.874324 237   
## [6] 0.01702866 0.5000000 6.165610 265   
## [7] 0.01580774 0.4949698 6.103582 246   
## [8] 0.01175941 0.4765625 5.876597 183   
## [9] 0.01497237 0.4623016 5.700743 233   
## [10] 0.01754273 0.4619289 5.696147 273   
## [11] 0.01728570 0.4551607 5.612687 269   
## [12] 0.01432978 0.3562300 5.633792 223

#Generating Rules - There are three parameters controlling the number of rules to be generated viz. Support and Confidence. Another parameter Lift is generated using Support and Confidence and is one of the major parameters to filter the generated rules. Support is an indication of how frequently the itemset appears in the dataset. Consider only the two transactions from the above output. The support of the item citrus fruit is 1/2 as it appears in only 1 out of the two transactions.Confidence is an indication of how often the rule has been found to be true. We will discuss more about confidence after generating the rules.  
  
library(RColorBrewer)  
  
plot ( arules1,control=list(  
 col = brewer.pal(11,"Spectral")  
),  
main="Association Rules Plot"  
)

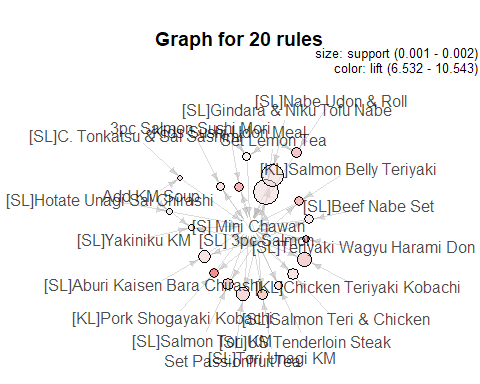
## To reduce overplotting, jitter is added! Use jitter = 0 to prevent jitter.



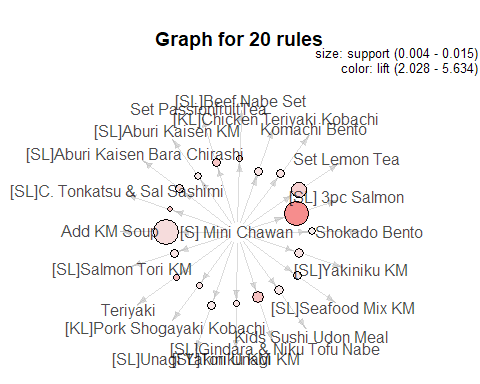
subrules1 <- head(sort(arules1, by="support"), 20)  
plot(subrules1, method = "graph", engine = "interactive")  
  
subrules2 <- head(sort(arules2, by="confidence"), 20)  
plot(subrules2, method = "graph", engine = "interactive")

rules\_df <- as(arules1,"data.frame")  
rules\_df$lhs\_support <- rules\_df$support/ rules\_df$confidence;  
rules\_df$rhs\_support <- rules\_df$confidence / rules\_df$lift;  
View(rules\_df)

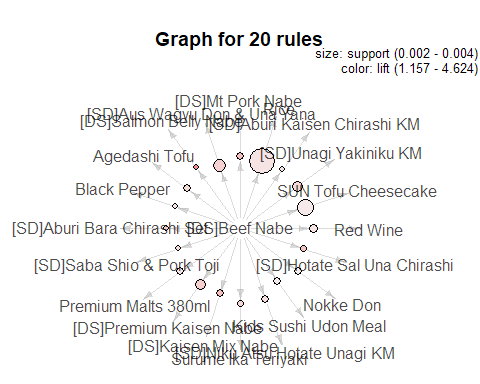
#What are customers likely to buy before buying whole milk? - default = LHS  
#What are customers likely to buy if they purchase whole milk? - default = RHS  
  
# Sort freq of top items  
menuqty <- table(data$Menu)  
menu100 <- as.data.frame(menuqty)  
menu50 <- head(menu100[order(menu100$Freq, decreasing= T),], n = 50)  
  
### support items: what they purchase before mini chawan - lhs  
rules.minichawan <- apriori (data=Agg.RTxn\_DD,   
 parameter=list (supp=0.001,conf = 0.01, minlen = 2),   
 appearance = list (default="lhs",rhs="[S] Mini Chawan"),   
 control = list (verbose=F))   
  
rules.minichawan <- head(sort(rules.minichawan, by="confidence"), 20)  
plot(rules.minichawan, method = "graph")



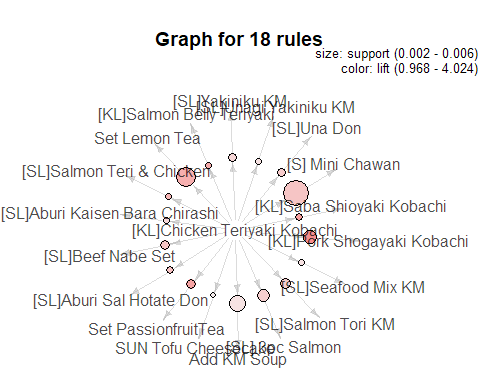
### support items: what they purchase after mini chawan - rhs  
rules.minichawan1 <- apriori (data=Agg.RTxn\_DD,   
 parameter=list (supp=0.001,conf = 0.01, minlen = 2),   
 appearance = list (default="rhs",lhs="[S] Mini Chawan"),   
 control = list (verbose=F))   
  
rules.minichawan1 <- head(sort(rules.minichawan1, by="confidence"), 20)  
plot(rules.minichawan1, method = "graph")



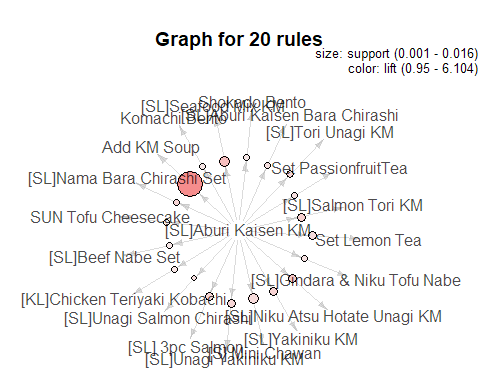
### support items: what they purchase after [DS]Beef Nabe - rhs  
rules.beefnabe <- apriori (data=Agg.RTxn\_DD,   
 parameter=list (supp=0.001,conf = 0.01, minlen = 2),   
 appearance = list (default="rhs",lhs="[DS]Beef Nabe"),   
 control = list (verbose=F))   
  
rules.beefnabe <- head(sort(rules.beefnabe, by="confidence"), 20)  
plot(rules.beefnabe, method = "graph")



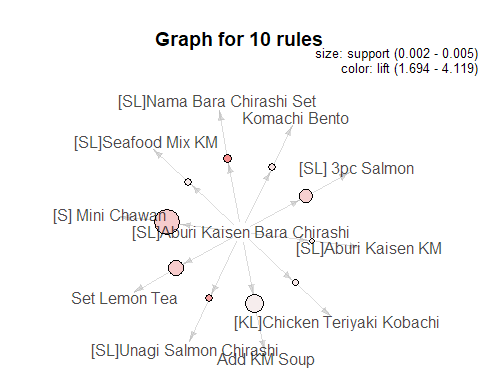
### support items: what they purchase after [KL]Chicken Teriyaki Kobachi- rhs  
rules.kobachi <- apriori (data=Agg.RTxn\_DD,   
 parameter=list (supp=0.001,conf = 0.05, minlen = 2),   
 appearance = list (default="rhs",lhs="[KL]Chicken Teriyaki Kobachi"),   
 control = list (verbose=F))   
  
rules.kobachi <- head(sort(rules.kobachi, by="confidence"), 20)  
plot(rules.kobachi, method = "graph")



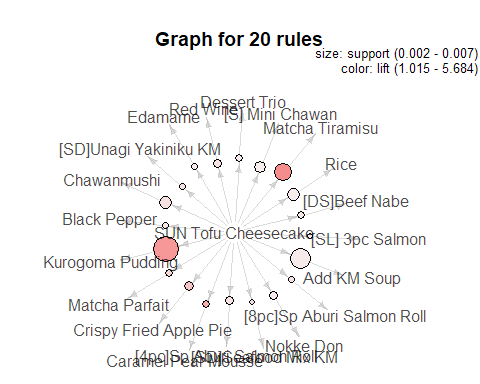
### support items: what they purchase after [SL]Aburi Kaisen KM rhs  
  
rules.kaisenkm <- apriori (data=Agg.RTxn\_DD,   
 parameter=list (supp=0.001,conf = 0.01, minlen = 2),   
 appearance = list (default="rhs",lhs="[SL]Aburi Kaisen KM"),   
 control = list (verbose=F))   
  
rules.kaisenkm <- head(sort(rules.kaisenkm, by="confidence"), 20)  
plot(rules.kaisenkm, method = "graph")



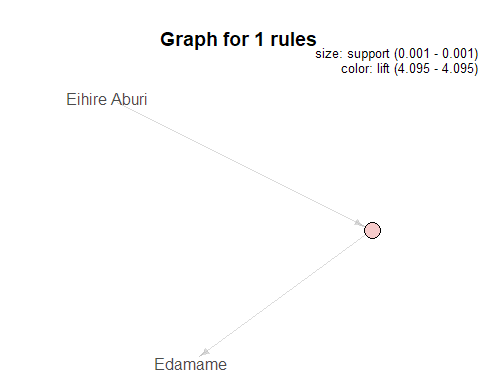
### support items: what they purchase after [SL]Aburi Kaisen Bara Chirashi- rhs  
  
rules.kaisenkm <- apriori (data=Agg.RTxn\_DD,   
 parameter=list (supp=0.001,conf = 0.01, minlen = 2),   
 appearance = list (default="rhs",lhs="[SL]Aburi Kaisen Bara Chirashi"),   
 control = list (verbose=F))   
  
rules.kaisenkm <- head(sort(rules.kaisenkm, by="confidence"), 10)  
plot(rules.kaisenkm, method = "graph")



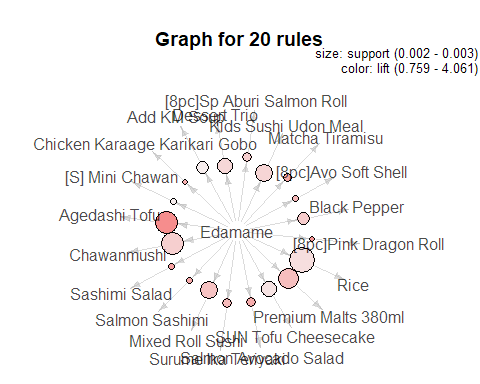
##-------------------------------------------  
### support items: get rules that lead to buying 'SUN Tofu Cheesecake'  
  
rules.tofu <- apriori (data=Agg.RTxn\_DD,   
 parameter=list (supp=0.001,conf = 0.01, minlen = 2),   
 appearance = list (default="rhs",lhs="SUN Tofu Cheesecake"),   
 control = list (verbose=F))   
  
rules.tofu1 <- head(sort(rules.tofu, by="support"), 20)  
plot(rules.tofu1, method = "graph")



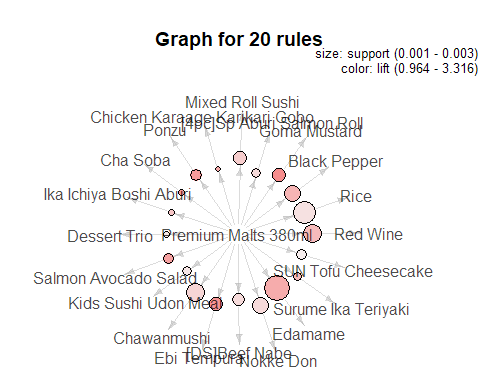
### support items: get rules that lead to buying 'Eihire Aburi'  
  
rules.eihire <- apriori (data=Agg.RTxn\_DD,   
 parameter=list (supp=0.001,conf = 0.01, minlen = 2),   
 appearance = list (default="rhs",lhs="Eihire Aburi"),   
 control = list (verbose=F))   
  
rules.eihire1 <- head(sort(rules.eihire, by="support"), 10)  
plot(rules.eihire1, method = "graph")



### support items: get rules that lead to buying 'Edamame'  
  
rules.edamame <- apriori (data=Agg.RTxn\_DD,   
 parameter=list (supp=0.001,conf = 0.01, minlen = 2),   
 appearance = list (default="rhs",lhs="Edamame"),   
 control = list (verbose=F))   
  
rules.edamame1 <- head(sort(rules.edamame, by="support"), 20)  
plot(rules.edamame1, method = "graph")



### support items: get rules that lead to buying 'Premium Malts 380ml'  
  
rules.beer <- apriori (data=Agg.RTxn\_DD,   
 parameter=list (supp=0.001,conf = 0.01, minlen = 2),   
 appearance = list (default="rhs",lhs="Premium Malts 380ml"),   
 control = list (verbose=F))   
  
rules.beer1 <- head(sort(rules.beer, by="support"), 20)  
plot(rules.beer1, method = "graph")



### support items: get rules that lead to buying 'Hamachi Kama Yaki'  
  
rules.hama <- apriori (data=Agg.RTxn\_DD,   
 parameter=list (supp=0.001,conf = 0.05, minlen = 2),   
 appearance = list (default="lhs",rhs="Hamachi Kama Yaki"),   
 control = list (verbose=F))   
  
rules.hama1 <- head(sort(rules.hama, by="support"), 20)  
plot(rules.hama1, method = "graph")

