Assignment Session 21-24

02 Nov 2019

Jagannath

# check data

Data Set

2. Perform the below given activities:

a. Apply PCA to the dataset and show proportion of variance

b. Perform PCA using SVD approach

c. Show the graphs of PCA components

library(readr)  
epi\_r <- read.csv  
View(epi\_r)  
data<-epi\_r  
View(data)  
head(data, n=10)

# data sets in package

data(package="arules")

# Split data

dt <- split(data$rating, data$arizona)

dt

# Loading arules package

require(arules)

require(arulesViz)

# Convert data to transaction level

dt2 = as(dt,"transactions")

dt2

summary(dt2)

inspect(dt2)

# Most Frequent Items

itemFrequency(dt2, type = "relative")

itemFrequencyPlot(dt2,topN = 5)

# with support parameters

itemFrequency(dt2, type = "relative")

itemFrequencyPlot(dt2,support= 0.10)

# aggregated data

rules = apriori(dt2, parameter=list(support=0.005, confidence=0.8))

rules = apriori(dt2, parameter=list(support=0.005, confidence=0.8, minlen = 3))

rules = apriori(dt2, parameter=list(support=0.005, confidence=0.8, maxlen = 4))

rules

summary(rules)

inspect(rules[1:10]) # to view first 10 rules

#Convert rules into data frame

rules3 = as(rules, "data.frame")

write(rules, "C:/Users/Jagannath/Documents/assignment data acadgild/assignmnet 21-24/rules2.csv", sep=",")

# Show only particular product rules

inspect( subset( rules, subset = rhs %pin% "0" )[1:10])

# Show the top 10 rules

options(digits=2)

inspect(rules[1:10])

# Get Summary Information

summary(rules)

plot(rules)

plot(rules, method = "graph", interactive = T)

# Sort by Lift

rules<-sort(rules, by="lift", decreasing=TRUE)

# Sort by Lift

rules<-sort(rules, by="lift", decreasing=TRUE)

# Remove Unnecessary Rules

subset.matrix <- is.subset(rules, rules)

subset.matrix[lower.tri(subset.matrix, diag=T)] <- NA

redundant <- colSums(subset.matrix, na.rm=T) >= 1

which(redundant)

rules.pruned <- rules[!redundant]

rules<-rules.pruned

rules

#Clean Rules

rules3$rules=gsub("\\{", "", rules3$rules)

rules3$rules=gsub("\\}", "", rules3$rules)

rules3$rules=gsub("\"", "", rules3$rules)

#Split the rule

library(splitstackshape)

Rules4=cSplit(rules3, "rules","=>")

names(Rules4)[names(Rules4) == 'rules\_1'] <- 'LHS'

Rules5=cSplit(Rules4, "LHS",",")

Rules6=subset(Rules5, select= -c(rules\_2))

names(Rules6)[names(Rules6) == 'rules\_3'] <- 'RHS'

# What are customers likely to buy before they purchase "Product A"

rules<-apriori(data=dt, parameter=list(supp=0.001,conf = 0.8),

appearance = list(default="lhs",rhs="0"),

control = list(verbose=F))

rules<-sort(rules, decreasing=TRUE,by="confidence")

inspect(rules[1:5])

# What are customers likely to buy if they purchased "Product A"

rules<-apriori(data=dt, parameter=list(supp=0.001,conf = 0.8),appearance = list(default="rhs",lhs="0"),control = list(verbose=F))

rules<-sort(rules, decreasing=TRUE,by="confidence")

inspect(rules[1:5])

rules

support<-seq(0.01,0.1,0.01)

support

rules\_count<-c(472,128,46,26,14, 10, 10,8,8,8)

rules\_count

plot(support,rules\_count,type = "l",main="Number of rules at different support %",col="darkred",lwd=3)

conf<-seq(0.10,1.0,0.10)

conf

rules\_count<-c(472,231,125,62,15,0,0,0,0,0)

rules\_count

plot(conf,rules\_count,type = "l",main="Number of rules at different confidence %",col="darkred",lwd=3)

#rules\_ec <- eclat(epi\_r, parameter = list(supp = 0.05))

#summary(rules\_ec)

#sorting out the most relevant rules

rules<-sort(rules, by="confidence", decreasing=TRUE)

inspect(rules[1:5])

rules<-sort(rules, by="lift", decreasing=TRUE)

inspect(rules[1:5])

########################################

library(factoextra)

library("factoextra")

data1<-na.exclude(data)

na.omit(data1)

data1.active <- data1[2:100, 2:6]

na.exclude(data1.active)

View(data1.active)

head(data1.active[, 2:5])

#Compute PCA in R using prcomp()

library(factoextra)

res.pca <- prcomp(data1.active, scale = TRUE)

res.pca

summary(res.pca)

fviz\_eig(res.pca)

fviz\_pca\_ind(res.pca, col.ind = "cos2", # Color by the quality of representation gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"), repel = TRUE # Avoid text overlapping)

fviz\_pca\_var(res.pca, col.var = "contrib", # Color by contributions to the PCgradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"), repel = TRUE # Avoid text overlapping)

fviz\_pca\_biplot(res.pca, repel = TRUE,col.var = "#2E9FDF", # Variables color col.ind = "#696969" # Individuals color)

library(factoextra)

# Eigenvalues

eig.val <- get\_eigenvalue(res.pca)

eig.val

# Results for Variables

res.var <- get\_pca\_var(res.pca)

res.var$coord # Coordinates

res.var$contrib # Contributions to the PCs

res.var$cos2 # Quality of representation

## title rating  
## 1 Lentil, Apple, and Turkey Wrap 2.500  
## 2 Boudin Blanc Terrine with Red Onion Confit 4.375  
## 3 Potato and Fennel Soup Hodge 3.750  
## 4 Mahi-Mahi in Tomato Olive Sauce 5.000  
## 5 Spinach Noodle Casserole 3.125  
## 6 The Best Blts 4.375  
## 7 Ham and Spring Vegetable Salad with Shallot Vinaigrette 4.375  
## 8 Spicy-Sweet Kumquats 3.750  
## 9 Korean Marinated Beef 4.375  
## 10 Ham Persillade with Mustard Potato Salad and Mashed Peas 3.750  
## calories protein fat sodium X.cakeweek X.wasteless X22.minute.meals  
## 1 426 30 7 559 0 0 0  
## 2 403 18 23 1439 0 0 0  
## 3 165 6 7 165 0 0 0  
## 4 NA NA NA NA 0 0 0  
## 5 547 20 32 452 0 0 0  
## 6 948 19 79 1042 0 0 0  
## 7 NA NA NA NA 0 0 0  
## 8 NA NA NA NA 0 0 0  
## 9 170 7 10 1272 0 0 0  
## 10 602 23 41 1696 0 0 0  
## X3.ingredient.recipes X30.days.of.groceries advance.prep.required  
## 1 0 0 0  
## 2 0 0 0  
## 3 0 0 0  
## 4 0 0 0  
## 5 0 0 0  
## 6 0 0 0  
## 7 0 0 0  
## 8 0 0 0  
## 9 0 0 0  
## 10 0 0 0  
## alabama alaska alcoholic almond amaretto anchovy anise anniversary  
## 1 0 0 0 0 0 0 0 0  
## 2 0 0 0 0 0 0 0 0  
## 3 0 0 0 0 0 0 0 0  
## 4 0 0 0 0 0 0 0 0  
## 5 0 0 0 0 0 0 0 0  
## 6 0 0 0 0 0 0 0 0  
## 7 0 0 0 0 0 0 0 0  
## 8 0 0 0 0 0 0 0 0  
## 9 0 0 0 0 0 0 0 0  
## 10 0 0 0 0 0 0 0 0  
## anthony.bourdain aperitif appetizer apple apple.juice apricot arizona  
## 1 0 0 0 1 0 0 0  
## 2 0 0 0 0 0 0 0  
## 3 0 0 0 0 0 0 0  
## 4 0 0 0 0 0 0 0  
## 5 0 0 0 0 0 0 0  
## 6 0 0 0 0 0 0 0  
## 7 0 0 0 0 0 0 0  
## 8 0 0 0 0 0 0 0  
## 9 0 0 0 0 0 0 0  
## 10 0 0 0 0 0 0 0  
## artichoke arugula asian.pear asparagus aspen atlanta australia avocado  
## 1 0 0 0 0 0 0 0 0  
## 2 0 0 0 0 0 0 0 0  
## 3 0 0 0 0 0 0 0 0  
## 4 0 0 0 0 0 0 0 0  
## 5 0 0 0 0 0 0 0 0  
## 6 0 0 0 0 0 0 0 0  
## 7 0 0 0 1 0 0 0 0  
## 8 0 0 0 0 0 0 0 0  
## 9 0 0 0 0 0 0 0 0  
  
## 9 0 0 0 0 0 0  
## 10 0 0 0 0 0 0

## title  
## 1 Lentil, Apple, and Turkey Wrap   
## 2 Boudin Blanc Terrine with Red Onion Confit   
## 3 Potato and Fennel Soup Hodge   
## 4 Mahi-Mahi in Tomato Olive Sauce   
## 5 Spinach Noodle Casserole   
## 6 The Best Blts   
## 7 Ham and Spring Vegetable Salad with Shallot Vinaigrette   
## 8 Spicy-Sweet Kumquats   
## 9 Korean Marinated Beef   
## 10 Ham Persillade with Mustard Potato Salad and Mashed Peas   
## 11 Yams Braised with Cream, Rosemary and Nutmeg   
## 12 Spicy Noodle Soup   
## 13 Banana-Chocolate Chip Cake With Peanut Butter Frosting   
## 14 Beef Tenderloin with Garlic and Brandy   
## 15 Peach Mustard   
## 16 Raw Cream of Spinach Soup   
## 17 Sweet Buttermilk Spoon Breads   
## 18 Crisp Braised Pork Shoulder   
## 19 Mozzarella-Topped Peppers with Tomatoes and Garlic   
## 20 Tuna, Asparagus, and New Potato Salad with Chive Vinaigrette and Fried Capers   
## 21 Asian Pear and Watercress Salad with Sesame Dressing   
## 22 "Fried" Chicken   
## 23 Fish Fillets in Parchment with Asparagus and Orange   
## 24 Pancetta and Taleggio Lasagna with Treviso   
## 25 Sea Salt-Roasted Pecans   
## 26 Garlic Baguette Crumbs   
## 27 Cucumber-Basil Egg Salad   
## 28 Dried Pear Crisps   
## 29 Green Bean, Red Onion, and Roast Potato Salad with Rosemary Vinaigrette   
## 30 Apricot-Cherry Shortcakes   
## 31 Asian Steak Topped with Bell Pepper Stir-Fry   
## 32 Moroccan-Style Preserved Lemons   
## 33 Roasted Sweet-Potato Spears with Bacon Vinaigrette   
## 34 Deviled Ham   
## 35 Fontina Mac with Squash and Sage   
## 36 Aztec Chicken   
## 37 Pastry Twists with Spiced Sugar-Honey Glaze   
## 38 Sauteed Broccoli Rabe   
## 39 Grouper with Tomato and Basil   
## 40 Better-Than-Pita Grill Bread   
## 41 Coconut-Key Lime Sheet Cake   
## 42 Baked Halibut with Orzo, Spinach, and Cherry Tomatoes   
## 43 Honey Rye   
## 44 Purple-Potato and Crab Gratin   
## 45 Grilled Beef, JÃ­cama, and Apple Salad   
## 46 Pickled Red Onions   
## 47 Spicy Black Beans and Rice   
## 48 Herbed Goat Cheese Spread with Mint   
## 49 Mexican Lime Soup   
## 50 Citrus Salad with Mint Sugar   
## 51 Mexican Chile and Mushroom Soup   
## 52 Peanut Butter-Banana Muffins   
## 53 Braised Chicken With Artichokes and Olives   
## 54 Pancetta Roast Chicken with Walnut Stuffing   
## 55 1977 Coconut Angel Food Cake   
## 56 Collard-and-Prosciutto Chicken Roulades Over Watercress Salad   
## 57 Veal Burgers Stuffed with Mozzarella Cheese   
## 58 Pumpkin Muffins   
## 59 Orange Balsamic Glaze   
## 60 Roasted Eggplant and Olive Spread with Pita Bread Chips   
## 61 Pecan Blue Cheese Crackers   
## 62 Romaine, Grilled Avocado, and Smoky Corn Salad with Chipotle-Caesar Dressing   
## 63 Southwest Corn Bread Stuffing with Corn and Green Chilies   
## 64 Colin Perryâ\200\231s Sorghum and Apple Sticky Pudding   
## 65 Mixed Berry Pavlovas   
## 66 Orange-Almond Cake with Chocolate Icing   
## 67 Scarborough Fair Tofu Burger   
## 68 Italian Vinaigrette   
## 69 White Chocolate Tartlets with Strawberries and Bananas   
## 70 Tomato-Infused Bulgur Pilaf with Fresh Basil   
## 71 Roasted Butternut Squash, Rosemary, and Garlic Lasagne   
## 72 Grilled Roast Beef and Stilton Sandwich   
## 73 Pear-Hazelnut Cheesecakes with Pear-Raspberry Sauce   
## 74 Nut Butter   
## 75 Cheese Ravioli with Fresh Tomato Sauce   
## 76 Banana Layer Cake with Cream Cheese Frosting   
## 77 South American-Style JÃ­cama and Orange Salad   
## 78 Roasted Acorn Squash and Chestnuts   
## 79 Maple Pumpkin Pots de CrÃ¨me   
## 80 Anadama Rolls with Mixed Seeds   
## 81 Braised Chicken and Rice with Orange, Saffron, Almond, and Pistachio Syrup   
## 82 Horseradish Dill Potato Salad   
## 83 Chicken in Green Pumpkin-Seed Sauce   
## 84 Jeweled Rice   
## 85 Braised Brisket with Bourbon-Peach Glaze   
## 86 Grilled Pork Chops with Classic Barbecue Sauce   
## 87 Bacon Crackers   
## 88 Roast Chicken With Sorghum and Squash   
## 89 Asparagus with Bacon and Onion   
## 90 Ricotta Omelets   
## 91 Carrot, Snow Pea, and Red Pepper Julienne in Honey Vinaigrette   
## 92 Salmon with Chili-Mango Salsa   
## 93 Turkey and Pinto Bean Chili   
## 94 Cucumber-Yogurt Salad with Mint   
## 95 Lamb Shanks Braised with Anise and Orange   
## 96 Parsley Mayo   
## 97 Acini di Pepe Pasta with Garlic and Olives   
## 98 Roast Beef Salad with Cabbage and Horseradish   
## 99 Savoy Cabbage and Arugula Salad   
## 100 Fennel, Beet and Orange Salad with Olives   
## 101 Shrimp Gazpacho   
## 102 Grilled Steak Salad with Beets and Scallions   
## 103 Parsnip and Apple Soup   
## 104 Short Rib Pot Pie   
## 105 Stout Floats   
## 106 Apricot-Pistachio Muffins Baked on the Grill   
## 107 Garlic Bruschetta   
## 108 Asian Noodles with Barbecued Duck Confit   
## 109 Sausage Fennel Stuffing   
## 110 Banana Split with Curried Chocolate-Coconut Sauce   
## 111 Escarole and Cheese Spoon Bread   
## 112 Honey-Ginger Barbecue Sauce   
## 113 Baked Pears with Rosemary, Gorgonzola Cheese and Port   
## 114 Kids' Matzoh Pizza   
## 115 Cranberry, Quince, and Pearl Onion Compote   
## 116 Chocolate-Mint Shamrock Shake   
## 117 Tropical Rum Punch   
## 118 Chickpea Salad Sandwich With Creamy Carrot-Radish Slaw   
## 119 Blackberry-Raspberry Sauce   
## 120 Laddie's Sub-Bourbon   
## 121 Red Cabbage and Onions   
## 122 Roast Cod with Potatoes, Onions, and Olives   
## 123 Spicy Tomato Sauce   
## 124 Cod Cannelloni with Swiss Chard and Roasted Pepper   
## 125 Swiss Chard with Roasted Pepper   
## 126 Chocolate Almond Butter   
## 127 Pastry Dough   
## 128 Roasted Bell Pepper Halves Stuffed with Bulgur and Spinach   
## 129 Spicy Sesame Noodles with Chopped Peanuts and Thai Basil   
## 130 Potato Gratin with Goat Cheese and Garlic   
## 131 Country Sausage and Sage Dressing   
## 132 Cherry Lime Virgin Rickeys   
## 133 Buttermilk-Spinach Spaetzle   
## 134 Radishes with Burrata   
## 135 Winter Squash SoufflÃ©   
## 136 Blueberry Streusel Cake   
## 137 Low-Fat Chicken Stock   
## 138 Honey Mustard Sauce   
## 139 Rosemary and Lemon Pinto Beans   
## 140 Asian Dipping Sauce   
## 141 Shrimp and Green Onion Pancakes   
## 142 Gnocchi with Tomato, Basil, and Olives   
## 143 Mustard-Ginger Shrimp Canapes   
## 144 Rumbrosia   
## 145 Roasted Root Vegetables   
## 146 Thai Vegetables   
## 147 Sage-Roasted Turkey with Caramelized Onions and Sage Gravy   
## rating calories protein fat sodium X.cakeweek X.wasteless  
## 1 2.500 426 30 7 559 0 0  
## 2 4.375 403 18 23 1439 0 0  
## 3 3.750 165 6 7 165 0 0  
## 4 5.000 NA NA NA NA 0 0  
## 5 3.125 547 20 32 452 0 0  
## 6 4.375 948 19 79 1042 0 0  
## 7 4.375 NA NA NA NA 0 0  
## 8 3.750 NA NA NA NA 0 0  
## 9 4.375 170 7 10 1272 0 0  
## 10 3.750 602 23 41 1696 0 0  
## 11 3.750 256 4 5 30 0 0  
## 12 4.375 NA NA NA NA 0 0  
## 13 4.375 766 12 48 439 0 0  
## 14 4.375 174 11 12 176 0 0  
## 15 3.125 134 4 3 1394 0 0  
## 16 4.375 382 5 31 977 0 0  
## 17 1.875 146 4 5 160 0 0  
## 18 4.375 890 59 68 1027 0 0  
## 19 5.000 107 5 7 344 0 0  
## 20 5.000 421 10 33 383 0 0  
## 21 4.375 345 11 19 423 0 0  
## 22 3.750 NA NA NA NA 0 0  
## 23 3.750 NA NA NA NA 0 0  
## 24 3.750 NA NA NA NA 0 0  
## 25 3.750 279 3 30 206 0 0  
## 26 0.000 95 1 7 103 0 0  
## 27 3.750 215 6 20 250 0 0  
## 28 2.500 14 0 0 0 0 0  
## 29 4.375 351 6 19 79 0 0  
## 30 4.375 311 5 5 226 0 0  
## 31 4.375 NA NA NA NA 0 0  
## 32 5.000 NA NA NA NA 0 0  
## 33 4.375 376 7 18 604 0 0  
## 34 3.125 185 10 13 765 0 0  
## 35 5.000 NA NA NA NA 0 0  
## 36 3.750 625 39 44 1248 0 0  
## 37 0.000 NA NA NA NA 0 0  
## 38 4.375 107 4 10 329 0 0  
## 39 4.375 336 44 16 413 0 0  
## 40 2.500 145 3 6 208 0 0  
## 41 4.375 483 5 35 100 0 0  
## 42 4.375 634 44 31 181 0 0  
## 43 0.000 NA NA NA NA 0 0  
## 44 3.750 NA NA NA NA 0 0  
## 45 4.375 NA NA NA NA 0 0  
## 46 4.375 90 2 0 881 0 0  
## 47 3.750 202 19 8 815 0 0  
##  
## 101 0 0 0 0 0 0 0 0  
## 102 0 0 0 0 1 0 0 0  
## 103 0 0 0 0 0 0 0 0  
## 104 0 0 0 0 1 0 0 0  
## 105 0 1 0 0 0 0 0 0  
## 106 0 0 0 0 0 0 0 0  
## 107 0 0 0 0 0 0 0 0  
## 108 0 0 0 0 0 0 0 0  
## 109 0 0 0 0 0 0 0 0  
## 110 0 1 0 0 0 0 0 0  
## 111 0 0 0 0 0 0 0 0  
## 112 0 0 0 0 0 0 0 0  
## 113 0 1 0 0 0 0 0 0  
## 114 0 0 0 0 0 0 0 0  
## 115 0 0 0 0 0 0 0 0  
## 116 0 0 0 0 0 0 0 0  
## 117 0 0 0 0 0 0 0 0  
## 118 0 0 0 0 0 0 0 0  
## 119 0 1 0 0 0 0 0 0  
## 120 0 0 0 0 0 0 0 0  
## 121 0 0 0 0 0 0 0 0  
## 122 0 0 0 0 0 0 0 0  
## 123 0 0 0 0 0 0 0 0  
## 124 0 0 0 0 0 0 0 0  
## 125 0 0 0 0 0 0 0 0  
## 126 0 0 0 0 0 0 0 0  
## 127 0 0 0 0 0 0 0 0  
## 128 0 0 0 0 0 0 0 0  
## 129 0 0 0 0 1 0 0 0  
## 130 0 0 0 0 0 0 0 0  
## 131 0 0 0 0 0 0 0 0  
## 132 0 0 0 0 0 0 0 0  
## 133 0 0 0 0 0 0 0 0  
## 134 0 0 0 0 0 0 0 0  
## 135 0 0 0 0 0 0 0 0  
## 136 0 1 0 0 0 0 0 0  
## 137 0 0 0 0 0 0 0 0  
## 138 0 0 0 0 0 0 0 0  
## 139 0 0 0 0 0 0 0 0  
## 140 0 0 0 0 0 0 0 0  
## 141 0 0 0 0 1 0 0 0  
## 142 0 0 0 0 1 0 0 0  
## 143 0 0 0 0 0 0 0 0  
## 144 0 0 0 0 0 0 0 0  
## 145 0 0 0 0 0 0 0 0  
## 146 0 0 0 0 0 0 0 0  
## 147 0 0 0 0 1 0 0 0  
## dorie.greenspan double.boiler dried.fruit drink drinks duck easter  
## 1 0 0 0 0 0 0 0  
## 2 0 0 1 0 0 0 0  
## 3 0 0 0 0 0 0 0  
## 4 0 0 0 0 0 0 0  
## 5 0 0 0 0 0 0 0  
## 6 0 0 0 0 0 0 0  
## 7 0 0 0 0 0 0 1  
## 8 0 0 0 0 0 0 0  
## 9 0 0 0 0 0 0 0  
## 10 0 0 0 0 0 0 0  
## 11 0 0 0 0 0 0 0  
## 12 0 0 0 0 0 0 0  
## 13 0 0 0 0 0 0 0  
## 14 0 0 0 0 0 0 0  
## 15 0 0 0 0 0 0 0  
## 16 0 0 0 0 0 0 0  
## 17 0 0 0 0 0 0 0  
## 18 0 0 0 0 0 0 0  
## 19 0 0 0 0 0 0 0  
## 20 0 0 0 0 0 0 0  
## 21 0 0 0 0 0 0 0  
## 116 0 0  
## 117 0 0  
## 118 0 0  
## 119 0 0  
## 120 0 0  
## 121 0 0  
## 122 0 0  
## 123 0 0  
## 124 0 0  
## 125 0 0  
## 126 0 0  
## 127 0 0  
## 128 0 0  
## 129 0 0  
## 130 0 0  
## 131 0 0  
## 132 0 0  
## 133 0 0  
## 134 0 0  
## 135 0 0  
## 136 0 0  
## 137 0 0  
## 138 0 0  
## 139 0 0  
## 140 0 0  
## 141 0 0  
## 142 0 0  
## 143 0 0  
## 144 0 0  
## 145 0 0  
## 146 0 0  
## 147 0 1  
## [ reached getOption("max.print") -- omitted 19905 rows ]

head(data, n=10)

## title rating  
## 1 Lentil, Apple, and Turkey Wrap 2.500  
## 2 Boudin Blanc Terrine with Red Onion Confit 4.375  
## 3 Potato and Fennel Soup Hodge 3.750  
## 4 Mahi-Mahi in Tomato Olive Sauce 5.000  
## 5 Spinach Noodle Casserole 3.125  
## 6 The Best Blts 4.375  
## 7 Ham and Spring Vegetable Salad with Shallot Vinaigrette 4.375  
## 8 Spicy-Sweet Kumquats 3.750  
## 9 Korean Marinated Beef 4.375  
## 10 Ham Persillade with Mustard Potato Salad and Mashed Peas 3.750  
## calories protein fat sodium X.cakeweek X.wasteless X22.minute.meals  
## 1 426 30 7 559 0 0 0  
## 2 403 18 23 1439 0 0 0  
## 3 165 6 7 165 0 0 0  
## 4 NA NA NA NA 0 0 0  
## 5 547 20 32 452 0 0 0  
## 6 948 19 79 1042 0 0 0  
## 7 NA NA NA NA 0 0 0  
## 8 NA NA NA NA 0 0 0  
## 9 170 7 10 1272 0 0 0  
## 10 602 23 41 1696 0 0 0  
## X3.ingredient.recipes X30.days.of.groceries advance.prep.required

## [12541] 4.375 4.375 3.125 4.375 3.750 5.000 4.375 4.375 4.375 3.750 4.375  
## [12552] 0.000 4.375 5.000 3.750 4.375 4.375 0.000 4.375 3.125 3.750 5.000  
## [12563] 3.750 3.750 4.375 4.375 5.000 2.500 3.125 5.000 3.750 3.125 4.375  
## [12574] 4.375 3.750 5.000 5.000 3.750 2.500 4.375 5.000 4.375 3.125 4.375  
## [12585] 3.750 4.375 5.000 4.375 4.375 4.375 2.500 3.750 3.125 4.375 5.000  
## [12596] 3.125 3.750 0.000 4.375 3.750 4.375 0.000 4.375 3.750 0.000 5.000  
## [12607] 4.375 4.375 1.250 4.375 0.000 3.750 3.750 4.375 0.000 0.000 0.000  
## [12618] 4.375 0.000 4.375 3.750 4.375 4.375 3.750 5.000 5.000 4.375 3.750  
## [12629] 3.750 3.750 3.750 4.375 3.750 0.000 3.750 3.750 0.000 4.375 4.375  
## [12640] 3.750 1.875 3.750 4.375 5.000 3.750 3.750 3.125 4.375 2.500 4.375  
## [12651] 3.750 3.750 4.375 3.125 3.750 4.375 3.750 4.375 2.500 4.375 4.375  
## [12662] 4.375 4.375 3.750 3.750 5.000 3.750 4.375 2.500 0.000 0.000 4.375  
## [12673] 3.125 5.000 5.000 4.375 5.000 4.375 4.375 4.375 3.750 0.000 0.000  
## [12684] 4.375 0.000 4.375 4.375 3.750 3.125 4.375 4.375 4.375 3.750 4.375  
## [12695] 4.375 4.375 4.375 5.000 4.375 0.000 4.375 0.000 4.375 5.000 4.375  
## [12706] 4.375 5.000 4.375 5.000 3.750 4.375 5.000 3.750 4.375 4.375 5.000  
## [12717] 3.750 1.250 4.375 4.375 5.000 4.375 3.750 4.375 3.750 3.750 5.000  
## [12728] 4.375 3.125 5.000 5.000 5.000 1.875 3.750 3.750 3.750 0.000 4.375  
## [12739] 4.375 4.375 3.750 3.750 3.125 1.250 4.375 4.375 3.750 4.375 3.750  
## [12750] 3.750 4.375 3.750 3.750 4.375 4.375 4.375 2.500 3.125 4.375 4.375  
## [12761] 5.000 4.375 4.375 5.000 0.000 5.000 4.375 3.750 4.375 4.375 3.125  
## [12772] 3.750 3.125 3.750 3.750 4.375 0.000 5.000 5.000 3.750 4.375 1.250  
## [12783] 4.375 4.375 3.750 3.750 4.375 4.375 3.750 4.375 3.125 4.375 5.000  
## [12794] 5.000 3.125 3.750 3.750 3.750 5.000 4.375 5.000 5.000 5.000 3.750  
## [12805] 2.500 5.000 0.000 3.125 3.750 4.375 4.375 4.375 5.000 4.375 4.375  
## [12816] 3.750 5.000 4.375 0.000 5.000 3.750 5.000 4.375 4.375 1.250 4.375  
## [12827] 4.375 4.375 5.000 4.375 3.750 4.375 4.375 5.000 4.375 5.000 3.750  
## [12838] 2.500 3.750 4.375 0.000 5.000 4.375 4.375 3.750 3.750 4.375 3.750  
## [12849] 4.375 4.375 4.375 3.750 0.000 0.000 4.375 3.750 3.750 3.750 5.000  
## [12860] 3.750 4.375 4.375 0.000 4.375 0.000 4.375 0.000 4.375 0.000 3.125  
## [12871] 4.375 5.000 4.375 3.750 5.000 4.375 3.125 3.750 2.500 4.375 5.000  
## [12882] 3.750 3.750 5.000 0.000 1.250 3.750 4.375 3.750 4.375 3.750 5.000  
## [12893] 3.750 4.375 3.125 3.750 4.375 3.750 4.375 4.375 4.375 3.750 0.000  
## [12904] 4.375 3.750 4.375 4.375 4.375 3.750 5.000 3.750 4.375 3.125 3.125  
## [12915] 0.000 5.000 4.375 0.000 3.750 4.375 3.750 4.375 4.375 0.000 3.750  
## [12926] 3.125 3.125 3.750 4.375 5.000 0.000 4.375 3.750 0.000 5.000 4.375  
## [12937] 2.500 4.375 3.750 4.375 4.375 4.375 0.000 4.375 3.750 4.375 3.750  
## [12948] 4.375 3.750 5.000 4.375 4.375 3.125 4.375 4.375 0.000 4.375 3.750  
## [12959] 0.000 3.750 4.375 3.125 3.750 0.000 4.375 5.000 3.125 5.000 3.750  
## [12970] 3.125 3.125 5.000 4.375 4.375 0.000 3.125 0.000 3.750 3.750 4.375  
## [12981] 5.000 4.375 3.750 4.375 5.000 4.375 3.125 3.750 5.000 3.750 0.000  
## [12992] 0.000 5.000 5.000 4.375 4.375 4.375 0.000 5.000 3.750 0.000 3.750  
## [13003] 0.000 0.000 4.375 4.375 4.375 4.375 3.750 4.375 0.000 5.000 4.375  
## [13014] 4.375 4.375 3.750 4.375 4.375 0.000 4.375 4.375 0.000 4.375 3.750  
## [13025] 4.375 3.125 0.000 4.375 3.750 3.750 0.000 5.000 3.750 0.000 5.000  
## [13036] 5.000 4.375 3.125 3.125 4.375 3.125 4.375 5.000 4.375 3.750 3.750  
## [13047] 5.000 0.000 3.750 3.750 5.000 4.375 4.375 3.750 3.750 4.375 0.000  
## [13058] 3.750 3.750 3.125 5.000 3.125 4.375 4.375 3.750 3.750 3.750 3.750  
## [13069] 4.375 3.750 4.375 3.125 0.000 5.000 0.000 4.375 4.375 0.000 4.375  
## [13080] 0.000 3.750 2.500 4.375 4.375 4.375 0.000 5.000 3.125 2.500 4.375  
## [13091] 3.125 3.750 3.750 0.000 4.375 3.750 0.000 5.000 3.750 3.750 4.375  
## [13102] 4.375 0.000 3.125 3.125 5.000 3.750 4.375 4.375 3.750 4.375 3.750  
## [13113] 3.750 3.750 3.750 4.375 4.375 4.375 4.375 5.000 3.750 3.750 3.750  
## [13124] 3.750 0.000 3.750 4.375 0.000 3.750 4.375 4.375 4.375 3.750 4.375  
## [13135] 0.000 4.375 0.000 3.750 5.000 4.375 3.125 4.375 0.000 4.375 4.375  
## [13146] 4.375 4.375 4.375 4.375 4.375 3.750 5.000 4.375 3.750 3.750 3.750  
## [13157] 5.000 5.000 0.000 3.750 4.375 5.000 4.375 4.375 3.125 4.375 3.125  
## [13168] 4.375 5.000 3.750 5.000 5.000 4.375 5.000 5.000 3.125 4.375 4.375  
## [13179] 2.500 0.000 2.500 4.375 4.375 4.375 4.375 4.375 4.375 4.375 4.375  
## [13190] 3.750 3.750 2.500 4.375 4.375 3.750 3.125 4.375 0.000 4.375 1.875  
## [13201] 4.375 4.375 4.375 5.000 3.750 4.375 4.375 4.375 3.125 5.000 3.750  
## [13212] 4.375 3.750 0.000 4.375 4.375 5.000 0.000 0.000 4.375 5.000 0.000  
## [13223] 4.375 0.000 4.375 3.125 0.000 3.750 3.750 4.375 2.500 4.375 4.375  
## [13234] 4.375 4.375 3.750 3.750 3.750 3.125 5.000 4.375 4.375 0.000 0.000  
## [13245] 3.750 3.125 4.375 4.375 4.375 3.750 3.125 3.125 0.000 4.375 4.375  
## [13256] 4.375 0.000 3.125 3.750 0.000 5.000 3.750 4.375 3.750 4.375 1.250  
## [13267] 0.000 5.000 4.375 4.375 3.750 4.375 2.500 3.125 3.750 4.375 4.375  
## [13278] 5.000 3.750 5.000 4.375 1.875 4.375 5.000 4.375 3.125 3.750 0.000  
## [13289] 3.750 4.375 4.375 3.125 5.000 4.375 4.375 3.125 3.125 4.375 4.375  
## [19900] 4.375 4.375 3.125 4.375 3.750 4.375 3.750 4.375 5.000 4.375 3.750  
## [19911] 4.375 4.375 0.000 4.375 3.125 3.750 4.375 5.000 3.750 4.375 4.375  
## [19922] 4.375 4.375 4.375 3.125 4.375 3.750 3.125 4.375 4.375 4.375 0.000  
## [19933] 0.000 3.750 3.750 3.750 3.125 5.000 0.000 4.375 4.375 1.250 0.000  
## [19944] 0.000 5.000 4.375 4.375 3.750 3.125 3.750 3.750 3.750 3.750 3.750  
## [19955] 4.375 4.375 5.000 0.000 0.000 4.375 0.000 3.750 4.375 4.375 3.750  
## [19966] 3.750 3.125 4.375 0.000 3.750 3.750 3.125 4.375 4.375 4.375 0.000  
## [19977] 5.000 4.375 4.375 3.750 5.000 4.375 3.750 4.375 4.375 3.750 3.750  
## [19988] 3.750 0.000 4.375 5.000 5.000 0.000 4.375 2.500 2.500 3.750 4.375  
## [19999] 0.000 4.375 0.000 3.750 5.000 5.000 3.750 3.750 4.375 4.375 3.125  
## [20010] 4.375 5.000 0.000 3.750 5.000 4.375 3.125 4.375 4.375 5.000 4.375  
## [20021] 3.750 3.750 3.750 5.000 4.375 5.000 4.375 3.750 5.000 0.000 3.125  
## [20032] 3.125 4.375 2.500 2.500 5.000 3.750 3.750 3.750 3.125 4.375 4.375  
## [20043] 4.375 4.375  
##   
## $`1`  
## [1] 3.750 3.750 4.375 4.375 3.750 4.375 4.375 5.000

# Loading arules package  
require(arules)

## Loading required package: arules

## Loading required package: Matrix

##   
## Attaching package: 'arules'

## The following objects are masked from 'package:base':  
##   
## abbreviate, write

require(arulesViz)

## Loading required package: arulesViz

## Loading required package: grid

# Convert data to transaction level  
dt2 = as(dt,"transactions")

## Warning in asMethod(object): removing duplicated items in transactions

dt2

## transactions in sparse format with  
## 2 transactions (rows) and  
## 8 items (columns)

summary(dt2)

## transactions as itemMatrix in sparse format with  
## 2 rows (elements/itemsets/transactions) and  
## 8 columns (items) and a density of 0.6875   
##   
## most frequent items:  
## 3.75 4.375 5 0 1.25 (Other)   
## 2 2 2 1 1 3   
##   
## element (itemset/transaction) length distribution:  
## sizes  
## 3 8   
## 1 1   
##   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 3.00 4.25 5.50 5.50 6.75 8.00   
##   
## includes extended item information - examples:  
## labels  
## 1 0  
## 2 1.25  
## 3 1.875  
##   
## includes extended transaction information - examples:  
## transactionID  
## 1 0  
## 2 1

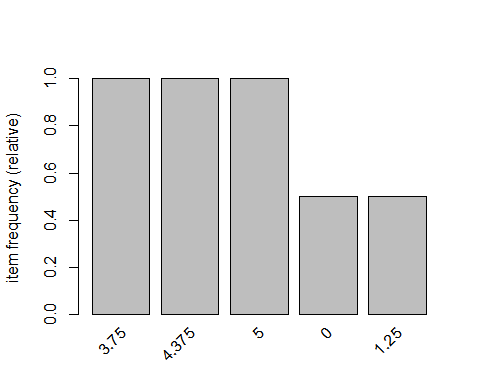
inspect(dt2)

## items transactionID  
## [1] {0,1.25,1.875,2.5,3.125,3.75,4.375,5} 0   
## [2] {3.75,4.375,5} 1

# Most Frequent Items  
itemFrequency(dt2, type = "relative")

## 0 1.25 1.875 2.5 3.125 3.75 4.375 5   
## 0.5 0.5 0.5 0.5 0.5 1.0 1.0 1.0

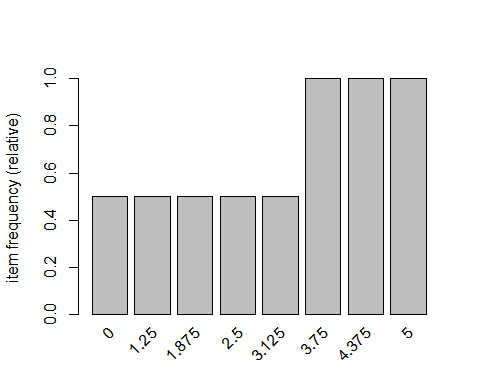
itemFrequencyPlot(dt2,topN = 5)



# with support parameters  
itemFrequency(dt2, type = "relative")

## 0 1.25 1.875 2.5 3.125 3.75 4.375 5   
## 0.5 0.5 0.5 0.5 0.5 1.0 1.0 1.0

itemFrequencyPlot(dt2,support= 0.10)



# aggregated data  
rules = apriori(dt2, parameter=list(support=0.005, confidence=0.8))

## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## 0.8 0.1 1 none FALSE TRUE 5 0.005 1  
## 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: 0   
##   
## set item appearances ...[0 item(s)] done [0.00s].  
## set transactions ...[8 item(s), 2 transaction(s)] done [0.00s].  
## sorting and recoding items ... [8 item(s)] done [0.00s].  
## creating transaction tree ... done [0.00s].  
## checking subsets of size 1 2 3 4 5 6 7 8 done [0.00s].  
## writing ... [984 rule(s)] done [0.00s].  
## creating S4 object ... done [0.00s].

rules = apriori(dt2, parameter=list(support=0.005, confidence=0.8, minlen = 3))

## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## 0.8 0.1 1 none FALSE TRUE 5 0.005 3  
## 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: 0   
##   
## set item appearances ...[0 item(s)] done [0.00s].  
## set transactions ...[8 item(s), 2 transaction(s)] done [0.00s].  
## sorting and recoding items ... [8 item(s)] done [0.00s].  
## creating transaction tree ... done [0.00s].  
## checking subsets of size 1 2 3 4 5 6 7 8 done [0.00s].  
## writing ... [940 rule(s)] done [0.00s].  
## creating S4 object ... done [0.00s].

rules = apriori(dt2, parameter=list(support=0.005, confidence=0.8, maxlen = 4))

## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## 0.8 0.1 1 none FALSE TRUE 5 0.005 1  
## maxlen target ext  
## 4 rules FALSE  
##   
## Algorithmic control:  
## filter tree heap memopt load sort verbose  
## 0.1 TRUE TRUE FALSE TRUE 2 TRUE  
##   
## Absolute minimum support count: 0   
##   
## set item appearances ...[0 item(s)] done [0.00s].  
## set transactions ...[8 item(s), 2 transaction(s)] done [0.00s].  
## sorting and recoding items ... [8 item(s)] done [0.00s].  
## creating transaction tree ... done [0.00s].  
## checking subsets of size 1 2 3 4

## Warning in apriori(dt2, parameter = list(support = 0.005, confidence =  
## 0.8, : Mining stopped (maxlen reached). Only patterns up to a length of 4  
## returned!

## done [0.00s].  
## writing ... [472 rule(s)] done [0.00s].  
## creating S4 object ... done [0.00s].

rules

## set of 472 rules

summary(rules)

## set of 472 rules  
##   
## rule length distribution (lhs + rhs):sizes  
## 1 2 3 4   
## 3 41 153 275   
##   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 3.000 4.000 3.483 4.000 4.000   
##   
## summary of quality measures:  
## support confidence lift count   
## Min. :0.5000 Min. :1 Min. :1.000 Min. :1.000   
## 1st Qu.:0.5000 1st Qu.:1 1st Qu.:1.000 1st Qu.:1.000   
## Median :0.5000 Median :1 Median :2.000 Median :1.000   
## Mean :0.5127 Mean :1 Mean :1.593 Mean :1.025   
## 3rd Qu.:0.5000 3rd Qu.:1 3rd Qu.:2.000 3rd Qu.:1.000   
## Max. :1.0000 Max. :1 Max. :2.000 Max. :2.000   
##   
## mining info:  
## data ntransactions support confidence  
## dt2 2 0.005 0.8

inspect(rules[1:10]) # to view first 10 rules

## lhs rhs support confidence lift count  
## [1] {} => {3.75} 1.0 1 1 2   
## [2] {} => {4.375} 1.0 1 1 2   
## [3] {} => {5} 1.0 1 1 2   
## [4] {0} => {1.25} 0.5 1 2 1   
## [5] {1.25} => {0} 0.5 1 2 1   
## [6] {0} => {1.875} 0.5 1 2 1   
## [7] {1.875} => {0} 0.5 1 2 1   
## [8] {0} => {2.5} 0.5 1 2 1   
## [9] {2.5} => {0} 0.5 1 2 1   
## [10] {0} => {3.125} 0.5 1 2 1

#Convert rules into data frame  
rules3 = as(rules, "data.frame")

write(rules, "C:/Users/Jagannath/Documents/assignment data acadgild/assignmnet 21-24/rules2.csv", sep=",")  
  
# Show only particular product rules  
inspect( subset( rules, subset = rhs %pin% "0" )[1:10])

## lhs rhs support confidence lift count  
## [1] {1.25} => {0} 0.5 1 2 1   
## [2] {1.875} => {0} 0.5 1 2 1   
## [3] {2.5} => {0} 0.5 1 2 1   
## [4] {3.125} => {0} 0.5 1 2 1   
## [5] {1.25,1.875} => {0} 0.5 1 2 1   
## [6] {1.25,2.5} => {0} 0.5 1 2 1   
## [7] {1.25,3.125} => {0} 0.5 1 2 1   
## [8] {1.25,3.75} => {0} 0.5 1 2 1   
## [9] {1.25,4.375} => {0} 0.5 1 2 1   
## [10] {1.25,5} => {0} 0.5 1 2 1

# Show the top 10 rules  
options(digits=2)  
inspect(rules[1:10])

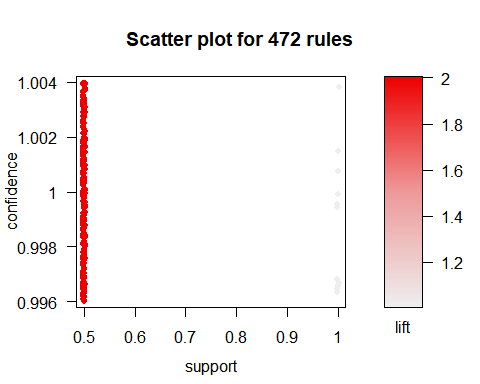
## lhs rhs support confidence lift count  
## [1] {} => {3.75} 1.0 1 1 2   
## [2] {} => {4.375} 1.0 1 1 2   
## [3] {} => {5} 1.0 1 1 2   
## [4] {0} => {1.25} 0.5 1 2 1   
## [5] {1.25} => {0} 0.5 1 2 1   
## [6] {0} => {1.875} 0.5 1 2 1   
## [7] {1.875} => {0} 0.5 1 2 1   
## [8] {0} => {2.5} 0.5 1 2 1   
## [9] {2.5} => {0} 0.5 1 2 1   
## [10] {0} => {3.125} 0.5 1 2 1

# Get Summary Information  
  
summary(rules)

## set of 472 rules  
##   
## rule length distribution (lhs + rhs):sizes  
## 1 2 3 4   
## 3 41 153 275   
##   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.0 3.0 4.0 3.5 4.0 4.0   
##   
## summary of quality measures:  
## support confidence lift count   
## Min. :0.50 Min. :1 Min. :1.00 Min. :1.00   
## 1st Qu.:0.50 1st Qu.:1 1st Qu.:1.00 1st Qu.:1.00   
## Median :0.50 Median :1 Median :2.00 Median :1.00   
## Mean :0.51 Mean :1 Mean :1.59 Mean :1.03   
## 3rd Qu.:0.50 3rd Qu.:1 3rd Qu.:2.00 3rd Qu.:1.00   
## Max. :1.00 Max. :1 Max. :2.00 Max. :2.00   
##   
## mining info:  
## data ntransactions support confidence  
## dt2 2 0.005 0.8

plot(rules)

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



plot(rules, method = "graph", interactive = T)

## Warning in plot.rules(rules, method = "graph", interactive = T): The  
## parameter interactive is deprecated. Use engine='interactive' instead.

## Warning: plot: Too many rules supplied. Only plotting the best 100 rules  
## using 'support' (change control parameter max if needed)

# Sort by Lift  
rules<-sort(rules, by="lift", decreasing=TRUE)  
  
# Sort by Lift  
rules<-sort(rules, by="lift", decreasing=TRUE)  
  
# Remove Unnecessary Rules  
subset.matrix <- is.subset(rules, rules)  
subset.matrix[lower.tri(subset.matrix, diag=T)] <- NA

## Warning in `[<-`(`\*tmp\*`, as.vector(i), value = NA): x[.] <- val: x is  
## "ngTMatrix", val not in {TRUE, FALSE} is coerced; NA |--> TRUE.

redundant <- colSums(subset.matrix, na.rm=T) >= 1  
which(redundant)

## {0,1.25} {0,1.25} {0,1.875}   
## 1 2 3   
## {0,1.875} {0,2.5} {0,2.5}   
## 4 5 6   
## {0,3.125} {0,3.125} {1.25,1.875}   
## 7 8 9   
## {1.25,1.875} {1.25,2.5} {1.25,2.5}   
## 10 11 12   
## {1.25,3.125} {1.25,3.125} {1.875,2.5}   
## 13 14 15   
## {1.875,2.5} {1.875,3.125} {1.875,3.125}   
## 16 17 18   
## {2.5,3.125} {2.5,3.125} {0,1.25,1.875}   
## 19 20 21   
## {0,1.25,1.875} {0,1.25,1.875} {0,1.25,2.5}   
## 22 23 24   
## {0,1.25,2.5} {0,1.25,2.5} {0,1.25,3.125}   
## 25 26 27   
## {0,1.25,3.125} {0,1.25,3.125} {0,1.25,3.75}   
## 28 29 30   
## {0,1.25,3.75} {0,1.25,4.375} {0,1.25,4.375}   
## 31 32 33   
## {0,1.25,5} {0,1.25,5} {0,1.875,2.5}   
## 34 35 36   
## {0,1.875,2.5} {0,1.875,2.5} {0,1.875,3.125}   
## 37 38 39   
## {0,1.875,3.125} {0,1.875,3.125} {0,1.875,3.75}   
## 40 41 42   
## {0,1.875,3.75} {0,1.875,4.375} {0,1.875,4.375}   
## 43 44 45   
## {0,1.875,5} {0,1.875,5} {0,2.5,3.125}   
## 46 47 48   
## {0,2.5,3.125} {0,2.5,3.125} {0,2.5,3.75}   
## 49 50 51   
## {0,2.5,3.75} {0,2.5,4.375} {0,2.5,4.375}   
## 52 53 54   
## {0,2.5,5} {0,2.5,5} {0,3.125,3.75}   
## 55 56 57   
## {0,3.125,3.75} {0,3.125,4.375} {0,3.125,4.375}   
## 58 59 60   
## {0,3.125,5} {0,3.125,5} {1.25,1.875,2.5}   
## 61 62 63   
## {1.25,1.875,2.5} {1.25,1.875,2.5} {1.25,1.875,3.125}   
## 64 65 66   
## {1.25,1.875,3.125} {1.25,1.875,3.125} {1.25,1.875,3.75}   
## 67 68 69   
## {1.25,1.875,3.75} {1.25,1.875,4.375} {1.25,1.875,4.375}   
## 70 71 72   
## {1.25,1.875,5} {1.25,1.875,5} {1.25,2.5,3.125}   
## 73 74 75   
## {1.25,2.5,3.125} {1.25,2.5,3.125} {1.25,2.5,3.75}   
## 76 77 78   
## {1.25,2.5,3.75} {1.25,2.5,4.375} {1.25,2.5,4.375}   
## 79 80 81   
## {1.25,2.5,5} {1.25,2.5,5} {1.25,3.125,3.75}   
## 82 83 84   
## {1.25,3.125,3.75} {1.25,3.125,4.375} {1.25,3.125,4.375}   
## 85 86 87   
## {1.25,3.125,5} {1.25,3.125,5} {1.875,2.5,3.125}   
## 88 89 90   
## {1.875,2.5,3.125} {1.875,2.5,3.125} {1.875,2.5,3.75}   
## 91 92 93   
## {1.875,2.5,3.75} {1.875,2.5,4.375} {1.875,2.5,4.375}   
## 94 95 96   
## {1.875,2.5,5} {1.875,2.5,5} {1.875,3.125,3.75}   
## 97 98 99   
## {1.875,3.125,3.75} {1.875,3.125,4.375} {1.875,3.125,4.375}   
## 100 101 102   
## {1.875,3.125,5} {1.875,3.125,5} {2.5,3.125,3.75}   
## 103 104 105   
## {2.5,3.125,3.75} {2.5,3.125,4.375} {2.5,3.125,4.375}   
## 106 107 108   
## {2.5,3.125,5} {2.5,3.125,5} {0,1.25,1.875,2.5}   
## 109 110 111   
## {0,1.25,1.875,2.5} {0,1.25,1.875,2.5} {0,1.25,1.875,2.5}   
## 112 113 114   
## {0,1.25,1.875,3.125} {0,1.25,1.875,3.125} {0,1.25,1.875,3.125}   
## 115 116 117   
## {0,1.25,1.875,3.125} {0,1.25,1.875,3.75} {0,1.25,1.875,3.75}   
## 118 119 120   
## {0,1.25,1.875,3.75} {0,1.25,1.875,4.375} {0,1.25,1.875,4.375}   
## 121 122 123   
## {0,1.25,1.875,4.375} {0,1.25,1.875,5} {0,1.25,1.875,5}   
## 124 125 126   
## {0,1.25,1.875,5} {0,1.25,2.5,3.125} {0,1.25,2.5,3.125}   
## 127 128 129   
## {0,1.25,2.5,3.125} {0,1.25,2.5,3.125} {0,1.25,2.5,3.75}   
## 130 131 132   
## {0,1.25,2.5,3.75} {0,1.25,2.5,3.75} {0,1.25,2.5,4.375}   
## 133 134 135   
## {0,1.25,2.5,4.375} {0,1.25,2.5,4.375} {0,1.25,2.5,5}   
## 136 137 138   
## {0,1.25,2.5,5} {0,1.25,2.5,5} {0,1.25,3.125,3.75}   
## 139 140 141   
## {0,1.25,3.125,3.75} {0,1.25,3.125,3.75} {0,1.25,3.125,4.375}   
## 142 143 144   
## {0,1.25,3.125,4.375} {0,1.25,3.125,4.375} {0,1.25,3.125,5}   
## 145 146 147   
## {0,1.25,3.125,5} {0,1.25,3.125,5} {0,1.25,3.75,4.375}   
## 148 149 150   
## {0,1.25,3.75,4.375} {0,1.25,3.75,5} {0,1.25,3.75,5}   
## 151 152 153   
## {0,1.25,4.375,5} {0,1.25,4.375,5} {0,1.875,2.5,3.125}   
## 154 155 156   
## {0,1.875,2.5,3.125} {0,1.875,2.5,3.125} {0,1.875,2.5,3.125}   
## 157 158 159   
## {0,1.875,2.5,3.75} {0,1.875,2.5,3.75} {0,1.875,2.5,3.75}   
## 160 161 162   
## {0,1.875,2.5,4.375} {0,1.875,2.5,4.375} {0,1.875,2.5,4.375}   
## 163 164 165   
## {0,1.875,2.5,5} {0,1.875,2.5,5} {0,1.875,2.5,5}   
## 166 167 168   
## {0,1.875,3.125,3.75} {0,1.875,3.125,3.75} {0,1.875,3.125,3.75}   
## 169 170 171   
## {0,1.875,3.125,4.375} {0,1.875,3.125,4.375} {0,1.875,3.125,4.375}   
## 172 173 174   
## {0,1.875,3.125,5} {0,1.875,3.125,5} {0,1.875,3.125,5}   
## 175 176 177   
## {0,1.875,3.75,4.375} {0,1.875,3.75,4.375} {0,1.875,3.75,5}   
## 178 179 180   
## {0,1.875,3.75,5} {0,1.875,4.375,5} {0,1.875,4.375,5}   
## 181 182 183   
## {0,2.5,3.125,3.75} {0,2.5,3.125,3.75} {0,2.5,3.125,3.75}   
## 184 185 186   
## {0,2.5,3.125,4.375} {0,2.5,3.125,4.375} {0,2.5,3.125,4.375}   
## 187 188 189   
## {0,2.5,3.125,5} {0,2.5,3.125,5} {0,2.5,3.125,5}   
## 190 191 192   
## {0,2.5,3.75,4.375} {0,2.5,3.75,4.375} {0,2.5,3.75,5}   
## 193 194 195   
## {0,2.5,3.75,5} {0,2.5,4.375,5} {0,2.5,4.375,5}   
## 196 197 198   
## {0,3.125,3.75,4.375} {0,3.125,3.75,4.375} {0,3.125,3.75,5}   
## 199 200 201   
## {0,3.125,3.75,5} {0,3.125,4.375,5} {0,3.125,4.375,5}   
## 202 203 204   
## {1.25,1.875,2.5,3.125} {1.25,1.875,2.5,3.125} {1.25,1.875,2.5,3.125}   
## 205 206 207   
## {1.25,1.875,2.5,3.125} {1.25,1.875,2.5,3.75} {1.25,1.875,2.5,3.75}   
## 208 209 210   
## {1.25,1.875,2.5,3.75} {1.25,1.875,2.5,4.375} {1.25,1.875,2.5,4.375}   
## 211 212 213   
## {1.25,1.875,2.5,4.375} {1.25,1.875,2.5,5} {1.25,1.875,2.5,5}   
## 214 215 216   
## {1.25,1.875,2.5,5} {1.25,1.875,3.125,3.75} {1.25,1.875,3.125,3.75}   
## 217 218 219   
## {1.25,1.875,3.125,3.75} {1.25,1.875,3.125,4.375} {1.25,1.875,3.125,4.375}   
## 220 221 222   
## {1.25,1.875,3.125,4.375} {1.25,1.875,3.125,5} {1.25,1.875,3.125,5}   
## 223 224 225   
## {1.25,1.875,3.125,5} {1.25,1.875,3.75,4.375} {1.25,1.875,3.75,4.375}   
## 226 227 228   
## {1.25,1.875,3.75,5} {1.25,1.875,3.75,5} {1.25,1.875,4.375,5}   
## 229 230 231   
## {1.25,1.875,4.375,5} {1.25,2.5,3.125,3.75} {1.25,2.5,3.125,3.75}   
## 232 233 234   
## {1.25,2.5,3.125,3.75} {1.25,2.5,3.125,4.375} {1.25,2.5,3.125,4.375}   
## 235 236 237   
## {1.25,2.5,3.125,4.375} {1.25,2.5,3.125,5} {1.25,2.5,3.125,5}   
## 238 239 240   
## {1.25,2.5,3.125,5} {1.25,2.5,3.75,4.375} {1.25,2.5,3.75,4.375}   
## 241 242 243   
## {1.25,2.5,3.75,5} {1.25,2.5,3.75,5} {1.25,2.5,4.375,5}   
## 244 245 246   
## {1.25,2.5,4.375,5} {1.25,3.125,3.75,4.375} {1.25,3.125,3.75,4.375}   
## 247 248 249   
## {1.25,3.125,3.75,5} {1.25,3.125,3.75,5} {1.25,3.125,4.375,5}   
## 250 251 252   
## {1.25,3.125,4.375,5} {1.875,2.5,3.125,3.75} {1.875,2.5,3.125,3.75}   
## 253 254 255   
## {1.875,2.5,3.125,3.75} {1.875,2.5,3.125,4.375} {1.875,2.5,3.125,4.375}   
## 256 257 258   
## {1.875,2.5,3.125,4.375} {1.875,2.5,3.125,5} {1.875,2.5,3.125,5}   
## 259 260 261   
## {1.875,2.5,3.125,5} {1.875,2.5,3.75,4.375} {1.875,2.5,3.75,4.375}   
## 262 263 264   
## {1.875,2.5,3.75,5} {1.875,2.5,3.75,5} {1.875,2.5,4.375,5}   
## 265 266 267   
## {1.875,2.5,4.375,5} {1.875,3.125,3.75,4.375} {1.875,3.125,3.75,4.375}   
## 268 269 270   
## {1.875,3.125,3.75,5} {1.875,3.125,3.75,5} {1.875,3.125,4.375,5}   
## 271 272 273   
## {1.875,3.125,4.375,5} {2.5,3.125,3.75,4.375} {2.5,3.125,3.75,4.375}   
## 274 275 276   
## {2.5,3.125,3.75,5} {2.5,3.125,3.75,5} {2.5,3.125,4.375,5}   
## 277 278 279   
## {2.5,3.125,4.375,5} {3.75} {4.375}   
## 280 281 282   
## {5} {0,3.75} {0,4.375}   
## 283 284 285   
## {0,5} {1.25,3.75} {1.25,4.375}   
## 286 287 288   
## {1.25,5} {1.875,3.75} {1.875,4.375}   
## 289 290 291   
## {1.875,5} {2.5,3.75} {2.5,4.375}   
## 292 293 294   
## {2.5,5} {3.125,3.75} {3.125,4.375}   
## 295 296 297   
## {3.125,5} {3.75,4.375} {3.75,4.375}   
## 298 299 300   
## {3.75,5} {3.75,5} {4.375,5}   
## 301 302 303   
## {4.375,5} {0,1.25,3.75} {0,1.25,4.375}   
## 304 305 306   
## {0,1.25,5} {0,1.875,3.75} {0,1.875,4.375}   
## 307 308 309   
## {0,1.875,5} {0,2.5,3.75} {0,2.5,4.375}   
## 310 311 312   
## {0,2.5,5} {0,3.125,3.75} {0,3.125,4.375}   
## 313 314 315   
## {0,3.125,5} {0,3.75,4.375} {0,3.75,4.375}   
## 316 317 318   
## {0,3.75,5} {0,3.75,5} {0,4.375,5}   
## 319 320 321   
## {0,4.375,5} {1.25,1.875,3.75} {1.25,1.875,4.375}   
## 322 323 324   
## {1.25,1.875,5} {1.25,2.5,3.75} {1.25,2.5,4.375}   
## 325 326 327   
## {1.25,2.5,5} {1.25,3.125,3.75} {1.25,3.125,4.375}   
## 328 329 330   
## {1.25,3.125,5} {1.25,3.75,4.375} {1.25,3.75,4.375}   
## 331 332 333   
## {1.25,3.75,5} {1.25,3.75,5} {1.25,4.375,5}   
## 334 335 336   
## {1.25,4.375,5} {1.875,2.5,3.75} {1.875,2.5,4.375}   
## 337 338 339   
## {1.875,2.5,5} {1.875,3.125,3.75} {1.875,3.125,4.375}   
## 340 341 342   
## {1.875,3.125,5} {1.875,3.75,4.375} {1.875,3.75,4.375}   
## 343 344 345   
## {1.875,3.75,5} {1.875,3.75,5} {1.875,4.375,5}   
## 346 347 348   
## {1.875,4.375,5} {2.5,3.125,3.75} {2.5,3.125,4.375}   
## 349 350 351   
## {2.5,3.125,5} {2.5,3.75,4.375} {2.5,3.75,4.375}   
## 352 353 354   
## {2.5,3.75,5} {2.5,3.75,5} {2.5,4.375,5}   
## 355 356 357   
## {2.5,4.375,5} {3.125,3.75,4.375} {3.125,3.75,4.375}   
## 358 359 360   
## {3.125,3.75,5} {3.125,3.75,5} {3.125,4.375,5}   
## 361 362 363   
## {3.125,4.375,5} {3.75,4.375,5} {3.75,4.375,5}   
## 364 365 366   
## {3.75,4.375,5} {0,1.25,1.875,3.75} {0,1.25,1.875,4.375}   
## 367 368 369   
## {0,1.25,1.875,5} {0,1.25,2.5,3.75} {0,1.25,2.5,4.375}   
## 370 371 372   
## {0,1.25,2.5,5} {0,1.25,3.125,3.75} {0,1.25,3.125,4.375}   
## 373 374 375   
## {0,1.25,3.125,5} {0,1.25,3.75,4.375} {0,1.25,3.75,4.375}   
## 376 377 378   
## {0,1.25,3.75,5} {0,1.25,3.75,5} {0,1.25,4.375,5}   
## 379 380 381   
## {0,1.25,4.375,5} {0,1.875,2.5,3.75} {0,1.875,2.5,4.375}   
## 382 383 384   
## {0,1.875,2.5,5} {0,1.875,3.125,3.75} {0,1.875,3.125,4.375}   
## 385 386 387   
## {0,1.875,3.125,5} {0,1.875,3.75,4.375} {0,1.875,3.75,4.375}   
## 388 389 390   
## {0,1.875,3.75,5} {0,1.875,3.75,5} {0,1.875,4.375,5}   
## 391 392 393   
## {0,1.875,4.375,5} {0,2.5,3.125,3.75} {0,2.5,3.125,4.375}   
## 394 395 396   
## {0,2.5,3.125,5} {0,2.5,3.75,4.375} {0,2.5,3.75,4.375}   
## 397 398 399   
## {0,2.5,3.75,5} {0,2.5,3.75,5} {0,2.5,4.375,5}   
## 400 401 402   
## {0,2.5,4.375,5} {0,3.125,3.75,4.375} {0,3.125,3.75,4.375}   
## 403 404 405   
## {0,3.125,3.75,5} {0,3.125,3.75,5} {0,3.125,4.375,5}   
## 406 407 408   
## {0,3.125,4.375,5} {0,3.75,4.375,5} {0,3.75,4.375,5}   
## 409 410 411   
## {0,3.75,4.375,5} {1.25,1.875,2.5,3.75} {1.25,1.875,2.5,4.375}   
## 412 413 414   
## {1.25,1.875,2.5,5} {1.25,1.875,3.125,3.75} {1.25,1.875,3.125,4.375}   
## 415 416 417   
## {1.25,1.875,3.125,5} {1.25,1.875,3.75,4.375} {1.25,1.875,3.75,4.375}   
## 418 419 420   
## {1.25,1.875,3.75,5} {1.25,1.875,3.75,5} {1.25,1.875,4.375,5}   
## 421 422 423   
## {1.25,1.875,4.375,5} {1.25,2.5,3.125,3.75} {1.25,2.5,3.125,4.375}   
## 424 425 426   
## {1.25,2.5,3.125,5} {1.25,2.5,3.75,4.375} {1.25,2.5,3.75,4.375}   
## 427 428 429   
## {1.25,2.5,3.75,5} {1.25,2.5,3.75,5} {1.25,2.5,4.375,5}   
## 430 431 432   
## {1.25,2.5,4.375,5} {1.25,3.125,3.75,4.375} {1.25,3.125,3.75,4.375}   
## 433 434 435   
## {1.25,3.125,3.75,5} {1.25,3.125,3.75,5} {1.25,3.125,4.375,5}   
## 436 437 438   
## {1.25,3.125,4.375,5} {1.25,3.75,4.375,5} {1.25,3.75,4.375,5}   
## 439 440 441   
## {1.25,3.75,4.375,5} {1.875,2.5,3.125,3.75} {1.875,2.5,3.125,4.375}   
## 442 443 444   
## {1.875,2.5,3.125,5} {1.875,2.5,3.75,4.375} {1.875,2.5,3.75,4.375}   
## 445 446 447   
## {1.875,2.5,3.75,5} {1.875,2.5,3.75,5} {1.875,2.5,4.375,5}   
## 448 449 450   
## {1.875,2.5,4.375,5} {1.875,3.125,3.75,4.375} {1.875,3.125,3.75,4.375}   
## 451 452 453   
## {1.875,3.125,3.75,5} {1.875,3.125,3.75,5} {1.875,3.125,4.375,5}   
## 454 455 456   
## {1.875,3.125,4.375,5} {1.875,3.75,4.375,5} {1.875,3.75,4.375,5}   
## 457 458 459   
## {1.875,3.75,4.375,5} {2.5,3.125,3.75,4.375} {2.5,3.125,3.75,4.375}   
## 460 461 462   
## {2.5,3.125,3.75,5} {2.5,3.125,3.75,5} {2.5,3.125,4.375,5}   
## 463 464 465   
## {2.5,3.125,4.375,5} {2.5,3.75,4.375,5} {2.5,3.75,4.375,5}   
## 466 467 468   
## {2.5,3.75,4.375,5} {3.125,3.75,4.375,5} {3.125,3.75,4.375,5}   
## 469 470 471   
## {3.125,3.75,4.375,5}   
## 472

rules.pruned <- rules[!redundant]  
rules<-rules.pruned  
rules

## set of 0 rules

#Clean Rules  
rules3$rules=gsub("\\{", "", rules3$rules)  
rules3$rules=gsub("\\}", "", rules3$rules)  
rules3$rules=gsub("\"", "", rules3$rules)  
  
#Split the rule  
library(splitstackshape)  
Rules4=cSplit(rules3, "rules","=>")  
names(Rules4)[names(Rules4) == 'rules\_1'] <- 'LHS'  
Rules5=cSplit(Rules4, "LHS",",")  
Rules6=subset(Rules5, select= -c(rules\_2))  
names(Rules6)[names(Rules6) == 'rules\_3'] <- 'RHS'  
  
# What are customers likely to buy before they purchase "Product A"  
rules<-apriori(data=dt, parameter=list(supp=0.001,conf = 0.8),   
 appearance = list(default="lhs",rhs="0"),  
 control = list(verbose=F))

## Warning in asMethod(object): removing duplicated items in transactions

rules<-sort(rules, decreasing=TRUE,by="confidence")  
inspect(rules[1:5])

## lhs rhs support confidence lift count  
## [1] {1.25} => {0} 0.5 1 2 1   
## [2] {1.875} => {0} 0.5 1 2 1   
## [3] {2.5} => {0} 0.5 1 2 1   
## [4] {3.125} => {0} 0.5 1 2 1   
## [5] {1.25,1.875} => {0} 0.5 1 2 1

# What are customers likely to buy if they purchased "Product A"  
rules<-apriori(data=dt, parameter=list(supp=0.001,conf = 0.8),   
 appearance = list(default="rhs",lhs="0"),  
 control = list(verbose=F))

## Warning in asMethod(object): removing duplicated items in transactions

rules<-sort(rules, decreasing=TRUE,by="confidence")  
inspect(rules[1:5])

## lhs rhs support confidence lift count  
## [1] {} => {3.75} 1.0 1 1 2   
## [2] {} => {4.375} 1.0 1 1 2   
## [3] {} => {5} 1.0 1 1 2   
## [4] {0} => {1.25} 0.5 1 2 1   
## [5] {0} => {1.875} 0.5 1 2 1

rules

## set of 10 rules

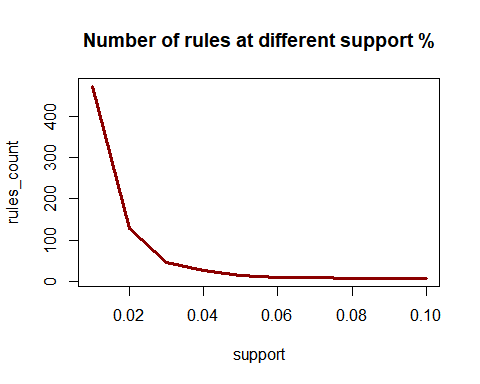
support<-seq(0.01,0.1,0.01)  
support

## [1] 0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.10

rules\_count<-c(472,128,46,26,14, 10, 10,8,8,8)  
rules\_count

## [1] 472 128 46 26 14 10 10 8 8 8

plot(support,rules\_count,type = "l",main="Number of rules at different support %",  
 col="darkred",lwd=3)

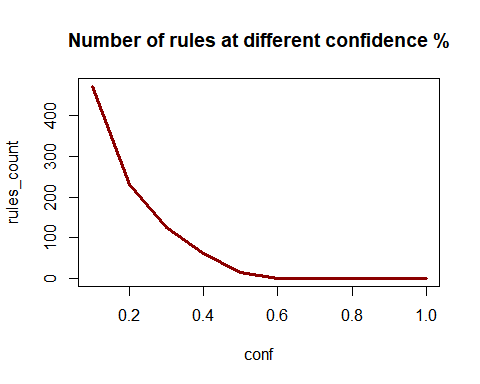


conf<-seq(0.10,1.0,0.10)  
conf

## [1] 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0

rules\_count<-c(472,231,125,62,15,0,0,0,0,0)  
rules\_count

## [1] 472 231 125 62 15 0 0 0 0 0

plot(conf,rules\_count,type = "l",main="Number of rules at different confidence %",  
 col="darkred",lwd=3)

#rules\_ec <- eclat(epi\_r, parameter = list(supp = 0.05))  
  
#summary(rules\_ec)  
  
#sorting out the most relevant rules  
rules<-sort(rules, by="confidence", decreasing=TRUE)  
inspect(rules[1:5])

## lhs rhs support confidence lift count  
## [1] {} => {3.75} 1.0 1 1 2   
## [2] {} => {4.375} 1.0 1 1 2   
## [3] {} => {5} 1.0 1 1 2   
## [4] {0} => {1.25} 0.5 1 2 1   
## [5] {0} => {1.875} 0.5 1 2 1

rules<-sort(rules, by="lift", decreasing=TRUE)  
inspect(rules[1:5])

## lhs rhs support confidence lift count  
## [1] {0} => {1.25} 0.5 1 2 1   
## [2] {0} => {1.875} 0.5 1 2 1   
## [3] {0} => {2.5} 0.5 1 2 1   
## [4] {0} => {3.125} 0.5 1 2 1   
## [5] {} => {3.75} 1.0 1 1 2

########################################  
library(factoextra)

## Loading required package: ggplot2

## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at https://goo.gl/13EFCZ

library("factoextra")  
data1<-na.exclude(data)  
na.omit(data1)

## title  
## 1 Lentil, Apple, and Turkey Wrap   
## 2 Boudin Blanc Terrine with Red Onion Confit   
## 3 Potato and Fennel Soup Hodge   
## 5 Spinach Noodle Casserole   
## 6 The Best Blts   
## 9 Korean Marinated Beef   
## 10 Ham Persillade with Mustard Potato Salad and Mashed Peas   
## 11 Yams Braised with Cream, Rosemary and Nutmeg   
## 13 Banana-Chocolate Chip Cake With Peanut Butter Frosting   
## 14 Beef Tenderloin with Garlic and Brandy   
## 15 Peach Mustard   
## 16 Raw Cream of Spinach Soup   
## 17 Sweet Buttermilk Spoon Breads   
## 18 Crisp Braised Pork Shoulder   
## 19 Mozzarella-Topped Peppers with Tomatoes and Garlic   
## 20 Tuna, Asparagus, and New Potato Salad with Chive Vinaigrette and Fried Capers   
## 21 Asian Pear and Watercress Salad with Sesame Dressing   
## 25 Sea Salt-Roasted Pecans   
## 26 Garlic Baguette Crumbs   
## 27 Cucumber-Basil Egg Salad   
## 28 Dried Pear Crisps   
## 29 Green Bean, Red Onion, and Roast Potato Salad with Rosemary Vinaigrette   
## 30 Apricot-Cherry Shortcakes   
## 33 Roasted Sweet-Potato Spears with Bacon Vinaigrette   
## 34 Deviled Ham   
## 36 Aztec Chicken   
## 38 Sauteed Broccoli Rabe   
## 39 Grouper with Tomato and Basil   
## 40 Better-Than-Pita Grill Bread   
## 41 Coconut-Key Lime Sheet Cake   
## 42 Baked Halibut with Orzo, Spinach, and Cherry Tomatoes   
## 46 Pickled Red Onions   
## 47 Spicy Black Beans and Rice   
## 49 Mexican Lime Soup   
## 50 Citrus Salad with Mint Sugar   
## 51 Mexican Chile and Mushroom Soup   
## 52 Peanut Butter-Banana Muffins   
## 54 Pancetta Roast Chicken with Walnut Stuffing   
## 55 1977 Coconut Angel Food Cake   
## 57 Veal Burgers Stuffed with Mozzarella Cheese   
## 58 Pumpkin Muffins   
## 59 Orange Balsamic Glaze   
## 60 Roasted Eggplant and Olive Spread with Pita Bread Chips   
## 61 Pecan Blue Cheese Crackers   
## 62 Romaine, Grilled Avocado, and Smoky Corn Salad with Chipotle-Caesar Dressing   
## 63 Southwest Corn Bread Stuffing with Corn and Green Chilies   
## 64 Colin Perryâ\200\231s Sorghum and Apple Sticky Pudding   
## 65 Mixed Berry Pavlovas   
## 67 Scarborough Fair Tofu Burger   
## 68 Italian Vinaigrette   
## 69 White Chocolate Tartlets with Strawberries and Bananas   
## 70 Tomato-Infused Bulgur Pilaf with Fresh Basil   
## 71 Roasted Butternut Squash, Rosemary, and Garlic Lasagne   
## 72 Grilled Roast Beef and Stilton Sandwich   
## 73 Pear-Hazelnut Cheesecakes with Pear-Raspberry Sauce   
## 74 Nut Butter   
## 75 Cheese Ravioli with Fresh Tomato Sauce   
## 76 Banana Layer Cake with Cream Cheese Frosting   
## 77 South American-Style JÃ­cama and Orange Salad   
## 78 Roasted Acorn Squash and Chestnuts   
## 79 Maple Pumpkin Pots de CrÃ¨me   
## 81 Braised Chicken and Rice with Orange, Saffron, Almond, and Pistachio Syrup   
## 82 Horseradish Dill Potato Salad   
## 83 Chicken in Green Pumpkin-Seed Sauce   
## 84 Jeweled Rice   
## 85 Braised Brisket with Bourbon-Peach Glaze   
## 86 Grilled Pork Chops with Classic Barbecue Sauce   
## 88 Roast Chicken With Sorghum and Squash   
## 89 Asparagus with Bacon and Onion   
## 92 Salmon with Chili-Mango Salsa   
## 93 Turkey and Pinto Bean Chili   
## 94 Cucumber-Yogurt Salad with Mint   
## 95 Lamb Shanks Braised with Anise and Orange   
## 96 Parsley Mayo   
## 97 Acini di Pepe Pasta with Garlic and Olives   
## 98 Roast Beef Salad with Cabbage and Horseradish   
## 99 Savoy Cabbage and Arugula Salad   
## 100 Fennel, Beet and Orange Salad with Olives   
## 101 Shrimp Gazpacho   
## 103 Parsnip and Apple Soup   
## 105 Stout Floats   
## 106 Apricot-Pistachio Muffins Baked on the Grill   
## 107 Garlic Bruschetta   
## 108 Asian Noodles with Barbecued Duck Confit   
## 110 Banana Split with Curried Chocolate-Coconut Sauce   
## 111 Escarole and Cheese Spoon Bread   
## 112 Honey-Ginger Barbecue Sauce   
## 114 Kids' Matzoh Pizza   
## 115 Cranberry, Quince, and Pearl Onion Compote   
## 117 Tropical Rum Punch   
## 118 Chickpea Salad Sandwich With Creamy Carrot-Radish Slaw   
## 119 Blackberry-Raspberry Sauce   
## 120 Laddie's Sub-Bourbon   
## 121 Red Cabbage and Onions   
## 122 Roast Cod with Potatoes, Onions, and Olives   
## 123 Spicy Tomato Sauce   
## 125 Swiss Chard with Roasted Pepper   
## 126 Chocolate Almond Butter   
## 127 Pastry Dough   
## 129 Spicy Sesame Noodles with Chopped Peanuts and Thai Basil   
## 130 Potato Gratin with Goat Cheese and Garlic   
## 131 Country Sausage and Sage Dressing   
## 133 Buttermilk-Spinach Spaetzle   
## 134 Radishes with Burrata   
## 135 Winter Squash SoufflÃ©   
## 136 Blueberry Streusel Cake   
## 137 Low-Fat Chicken Stock   
## 138 Honey Mustard Sauce   
## 139 Rosemary and Lemon Pinto Beans   
## 140 Asian Dipping Sauce   
## 141 Shrimp and Green Onion Pancakes   
## 143 Mustard-Ginger Shrimp Canapes   
## 144 Rumbrosia   
## 146 Thai Vegetables   
## 147 Sage-Roasted Turkey with Caramelized Onions and Sage Gravy   
## 148 Shrimp Cakes with Andouille Sausage   
## 149 Creamy Tofu Salad   
## 150 Chocolate-Cherry Ice Cream Pie with Hot Fudge Sauce   
## 151 JalapeÃ±o-Cheddar Frittata   
## 152 Roasted Beets and Citrus with Feta   
## 153 Green Beans with Crisp Shallots, Chile, and Mint   
## 155 Cranberry Pear Tart with Gingerbread Crust   
## 156 Sauteed Veal with Shrimp, Mushroom, and Brandy Cream Sauce   
## 157 Lemon Vinaigrette   
## 159 Cranberry, Shallot, and Dried-Cherry Compote   
## 161 Peanut Butter Cream Tart   
## 162 Cheddar Chicken Tenders with Wilted Spinach   
## 163 Blueberry Cheesecake   
## 165 Apple Pie with Whisky-Soaked Cherries   
## 166 Parsleyed Yellow-Potato Salad   
## 167 Sauteed Fennel and Carrots   
## 168 Grilled Garlic-Marinated Skirt Steak with Lime   
## 169 Miniature Crab Cakes with Tomato Ginger Jam   
## 170 Egg Sandwich with Green Bean Slaw   
## 171 Red Wine Brasato with Glazed Root Vegetables   
## 173 Egg Salad with Lemon and Fennel   
## 175 Shaved Brussels Sprout and Shallot SautÃ©   
## 177 Roasted Carrot and Beet Salad with Feta   
## 178 Cassata Cake   
## 179 Baked Beans with Slab Bacon and Breadcrumbs   
## 180 Grilled Corn with Lime-Cilantro Butter   
## 181 Roasted Winter Squash and Parsnips with Maple Syrup Glaze and Marcona Almonds   
## 185 Roasted Bell Peppers with Basil and Balsamic Vinegar   
## 186 Homemade Tomato Ketchup   
## 188 Char-Grilled Beef Tenderloin with Three-Herb Chimichurri   
## 189 Pork Roast Braised with Milk and Fresh Herbs (Maiale al Latte )   
## 191 Chocolate Pecan Banana Tarts   
## rating calories protein fat sodium X.cakeweek X.wasteless  
## 1 2.5 4.3e+02 30 7 5.6e+02 0 0  
## 2 4.4 4.0e+02 18 23 1.4e+03 0 0  
## 3 3.8 1.6e+02 6 7 1.6e+02 0 0  
## 5 3.1 5.5e+02 20 32 4.5e+02 0 0  
## 6 4.4 9.5e+02 19 79 1.0e+03 0 0  
## 9 4.4 1.7e+02 7 10 1.3e+03 0 0  
## 10 3.8 6.0e+02 23 41 1.7e+03 0 0  
## 11 3.8 2.6e+02 4 5 3.0e+01 0 0  
## 13 4.4 7.7e+02 12 48 4.4e+02 0 0  
## 14 4.4 1.7e+02 11 12 1.8e+02 0 0  
## 15 3.1 1.3e+02 4 3 1.4e+03 0 0  
## 16 4.4 3.8e+02 5 31 9.8e+02 0 0  
## 17 1.9 1.5e+02 4 5 1.6e+02 0 0  
## 18 4.4 8.9e+02 59 68 1.0e+03 0 0  
## 19 5.0 1.1e+02 5 7 3.4e+02 0 0  
## 20 5.0 4.2e+02 10 33 3.8e+02 0 0  
## 21 4.4 3.4e+02 11 19 4.2e+02 0 0  
## 25 3.8 2.8e+02 3 30 2.1e+02 0 0  
## 26 0.0 9.5e+01 1 7 1.0e+02 0 0  
## 27 3.8 2.2e+02 6 20 2.5e+02 0 0  
## 28 2.5 1.4e+01 0 0 0.0e+00 0 0  
## 29 4.4 3.5e+02 6 19 7.9e+01 0 0  
## 30 4.4 3.1e+02 5 5 2.3e+02 0 0  
## 33 4.4 3.8e+02 7 18 6.0e+02 0 0  
## 34 3.1 1.8e+02 10 13 7.6e+02 0 0  
## 36 3.8 6.2e+02 39 44 1.2e+03 0 0  
## 38 4.4 1.1e+02 4 10 3.3e+02 0 0  
## 39 4.4 3.4e+02 44 16 4.1e+02 0 0  
## 40 2.5 1.4e+02 3 6 2.1e+02 0 0  
## 41 4.4 4.8e+02 5 35 1.0e+02 0 0  
## 42 4.4 6.3e+02 44 31 1.8e+02 0 0  
## 46 4.4 9.0e+01 2 0 8.8e+02 0 0  
## 47 3.8 2.0e+02 19 8 8.2e+02 0 0  
## 49 4.4 3.4e+02 14 21 1.7e+02 0 0  
## 50 4.4 1.9e+02 3 1 4.0e+00 0 0  
## 51 3.1 1.7e+02 8 12 5.1e+02 0 0  
## 52 3.8 2.8e+02 6 13 2.4e+02 0 0  
## 54 5.0 1.2e+03 89 87 5.8e+02 0 0  
## 55 3.8 2.7e+02 4 7 1.5e+02 0 0  
## 57 4.4 9.0e+02 38 70 1.4e+03 0 0  
## 58 4.4 2.2e+02 4 10 2.1e+02 0 0  
## 59 3.8 1.9e+02 2 3 7.0e+02 0 0  
## 60 3.8 1.8e+02 5 7 1.2e+02 0 0  
## 61 3.8 7.0e+01 2 6 6.0e+01 0 0  
  
## 127 0 0 0 0 0 0  
## 129 0 0 0 0 0 0  
## 130 0 0 0 0 0 0  
## 131 0 0 0 0 0 0  
## 133 0 0 0 0 0 0  
## 134 0 0 0 0 0 0  
1 0 0 0 0 0 0 0  
## 62 0 0 0 0 0 0 0  
## 63 0 0 0 0 0 0 0  
## 64 0 0 0 0 0 0 0  
## 65 0 0 0 0 0 0 0  
## 67 0 0 0 0 0 0 0  
## 68 0 0 0 0 0 0 0  
## 69 0 0 0 0 0 0 0  
## 70 0 0 0 0 0 0 0  
## 71 0 0 0 0 0 0 0  
## 72 0 0 0 0 0 0 0  
## 73 0 0 0 0 0 0 0  
## 74 0 0 0 0 0 0 0  
## 75 0 0 0 0 0 0 0  
## 76 0 0 0 0 0 0 0  
## 77 0 0 0 0 0 0 0  
## 78 0 0 0 0 0 0 0  
## 79 0 0 0 0 0 0 0  
## 81 1 0 0 0 0 0 0  
## 82 0 0 0 0 0 0 0  
## 83 0 0 0 0 0 0 0  
## 84 1 0 0 0 0 0 0  
## 85 0 0 0 0 0 0 0  
## 86 0 0 0 0 0 0 0  
## 88 0 0 0 0 0 0 0  
## 89 0 0 0 0 0 0 0  
## 92 0 0 0 0 0 0 0  
## 93 0 0 0 0 0 0 0  
## 94 0 0 0 0 0 0 0  
## 95 0 0 0 0 0 0 0  
## 96 0 0 0 0 0 0 0  
## 97 0 0 0 0 0 0 0  
## 98 0 0 0 0 0 0 0  
## 99 0 0 0 0 0 0 0  
## 100 0 0 0 0 0 0 0  
## 101 0 0 0 0 0 0 0  
## 103 0 0 0 0 0 0 0  
## 105 0 0 0 0 0 0 0  
## 106 1 0 0 0 0 0 0  
## 107 0 0 0 0 0 0 0  
## 108 0 0 0 0 0 0 0  
## 110 0 0 0 0 0 0 0  
## 111 0 0 0 0 0 0 0  
## 112 0 0 0 0 0 0 0  
## 114 0 0 0 0 0 0 0  
  
## 191 0 0  
## [ reached getOption("max.print") -- omitted 15717 rows ]

data1.active <- data1[2:100, 2:6]  
na.exclude(data1.active)

## rating calories protein fat sodium  
## 2 4.4 403 18 23 1439  
## 3 3.8 165 6 7 165  
## 5 3.1 547 20 32 452  
## 6 4.4 948 19 79 1042  
## 9 4.4 170 7 10 1272  
## 10 3.8 602 23 41 1696  
## 11 3.8 256 4 5 30  
## 13 4.4 766 12 48 439  
## 14 4.4 174 11 12 176  
## 15 3.1 134 4 3 1394  
## 16 4.4 382 5 31 977  
## 17 1.9 146 4 5 160  
## 18 4.4 890 59 68 1027  
## 19 5.0 107 5 7 344  
## 20 5.0 421 10 33 383  
## 21 4.4 345 11 19 423  
## 25 3.8 279 3 30 206  
## 26 0.0 95 1 7 103  
## 27 3.8 215 6 20 250  
## 28 2.5 14 0 0 0  
## 29 4.4 351 6 19 79  
## 30 4.4 311 5 5 226  
## 33 4.4 376 7 18 604  
## 34 3.1 185 10 13 765  
## 36 3.8 625 39 44 1248  
## 38 4.4 107 4 10 329  
## 39 4.4 336 44 16 413  
## 40 2.5 145 3 6 208  
## 41 4.4 483 5 35 100  
## 42 4.4 634 44 31 181  
## 46 4.4 90 2 0 881  
## 47 3.8 202 19 8 815  
## 49 4.4 338 14 21 174  
## 50 4.4 191 3 1 4  
## 51 3.1 166 8 12 508  
## 52 3.8 275 6 13 242  
## 54 5.0 1203 89 87 583  
## 55 3.8 266 4 7 148  
## 57 4.4 904 38 70 1413  
## 58 4.4 223 4 10 211  
## 59 3.8 194 2 3 697  
## 60 3.8 177 5 7 116  
## 61 3.8 70 2 6 60  
## 62 4.4 368 6 32 112  
## 63 5.0 293 7 15 565  
## 64 0.0 523 8 19 694  
## 65 3.8 252 4 7 89  
## 67 3.1 224 21 12 340  
## 68 3.8 185 0 20 155  
## 69 4.4 830 9 59 148  
## 70 0.0 195 7 5 469  
## 71 4.4 684 21 42 637  
## 72 4.4 641 39 37 907  
## 73 3.8 538 8 36 231  
## 74 3.8 264 7 24 84  
## 75 3.8 365 19 14 599  
## 76 4.4 926 9 56 569  
## 77 3.1 26 1 0 364  
## 78 3.8 230 2 7 9  
## 79 4.4 157 3 8 51  
## 81 4.4 1172 54 73 220  
## 82 4.4 298 6 12 199  
## 83 4.4 682 36 57 909  
## 84 5.0 517 7 18 20  
## 85 4.4 856 45 54 1797  
## 86 4.4 599 48 28 1038  
## 88 0.0 1143 63 77 311  
## 89 4.4 129 4 11 146  
## 92 4.4 571 36 37 106  
## 93 4.4 508 45 17 826  
## 94 3.8 62 2 3 603  
## 95 3.1 1118 92 70 1226  
## 96 0.0 306 0 34 302  
## 97 4.4 209 5 8 77  
## 98 3.1 645 22 52 324  
## 99 3.1 126 3 9 46  
## 100 4.4 133 2 8 91  
## 101 4.4 285 19 7 635  
## 103 4.4 310 3 25 89  
## 105 3.1 248 3 10 73  
## 106 5.0 247 5 10 185  
## 107 3.8 201 4 11 210  
## 108 5.0 519 14 25 1237  
## 110 3.8 1076 11 73 150  
## 111 4.4 338 15 18 515  
## 112 4.4 298 1 0 1430  
## 114 3.1 280 13 18 587  
## 115 3.8 171 1 0 6  
## 117 5.0 230 1 0 26  
## 118 3.8 959 21 60 1541  
## 119 4.4 129 1 0 2  
## 120 2.5 107 0 0 0  
## 121 3.8 86 3 1 339  
## 122 3.8 679 55 36 333  
## 123 0.0 112 2 7 12  
## 125 3.8 123 5 7 605  
## 126 0.0 273 2 28 5  
## 127 5.0 234 3 16 99  
## 129 4.4 724 21 28 1130

View(data1.active)  
head(data1.active[, 2:5])

## calories protein fat sodium  
## 2 403 18 23 1439  
## 3 165 6 7 165  
## 5 547 20 32 452  
## 6 948 19 79 1042  
## 9 170 7 10 1272  
## 10 602 23 41 1696

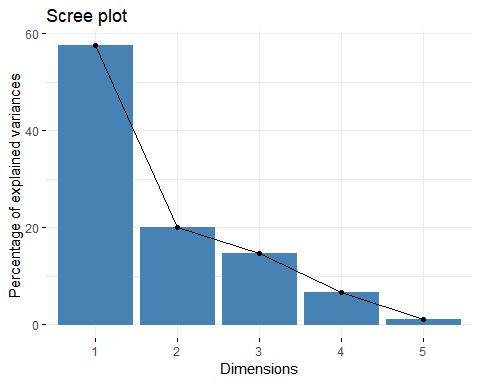
#Compute PCA in R using prcomp()  
library(factoextra)  
res.pca <- prcomp(data1.active, scale = TRUE)  
res.pca

## Standard deviations (1, .., p=5):  
## [1] 1.70 1.00 0.86 0.58 0.22  
##   
## Rotation (n x k) = (5 x 5):  
## PC1 PC2 PC3 PC4 PC5  
## rating -0.11 0.943 0.311 0.0031 -0.038  
## calories -0.56 -0.105 0.208 -0.2732 0.746  
## protein -0.51 -0.085 0.052 0.8484 -0.099  
## fat -0.54 -0.169 0.241 -0.4344 -0.657  
## sodium -0.35 0.253 -0.894 -0.1299 -0.023

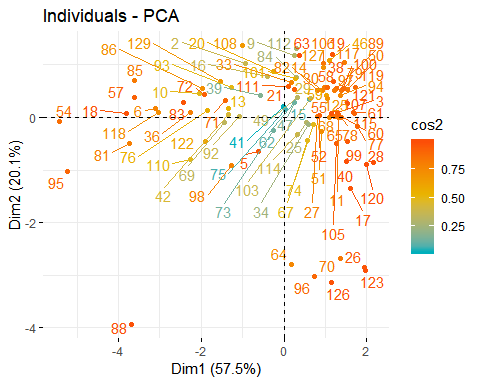
summary(res.pca)

## Importance of components:  
## PC1 PC2 PC3 PC4 PC5  
## Standard deviation 1.696 1.002 0.856 0.579 0.2243  
## Proportion of Variance 0.575 0.201 0.147 0.067 0.0101  
## Cumulative Proportion 0.575 0.776 0.923 0.990 1.0000

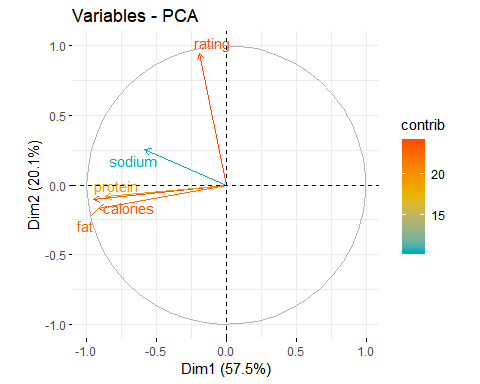
fviz\_eig(res.pca)



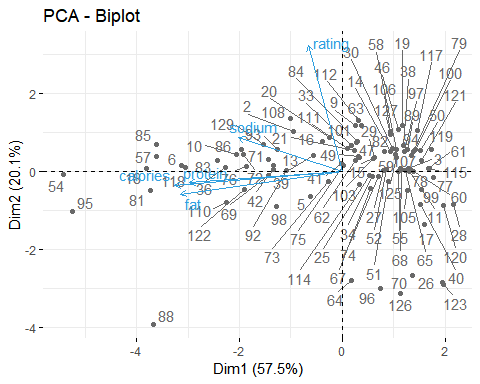
fviz\_pca\_ind(res.pca,  
 col.ind = "cos2", # Color by the quality of representation  
 gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),  
 repel = TRUE # Avoid text overlapping  
)



fviz\_pca\_var(res.pca,  
 col.var = "contrib", # Color by contributions to the PC  
 gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),  
 repel = TRUE # Avoid text overlapping  
)



fviz\_pca\_biplot(res.pca, repel = TRUE,  
 col.var = "#2E9FDF", # Variables color  
 col.ind = "#696969" # Individuals color  
)



library(factoextra)  
# Eigenvalues  
eig.val <- get\_eigenvalue(res.pca)  
eig.val

## eigenvalue variance.percent cumulative.variance.percent  
## Dim.1 2.88 57.5 58  
## Dim.2 1.00 20.1 78  
## Dim.3 0.73 14.7 92  
## Dim.4 0.33 6.7 99  
## Dim.5 0.05 1.0 100

# Results for Variables  
res.var <- get\_pca\_var(res.pca)  
res.var$coord # Coordinates

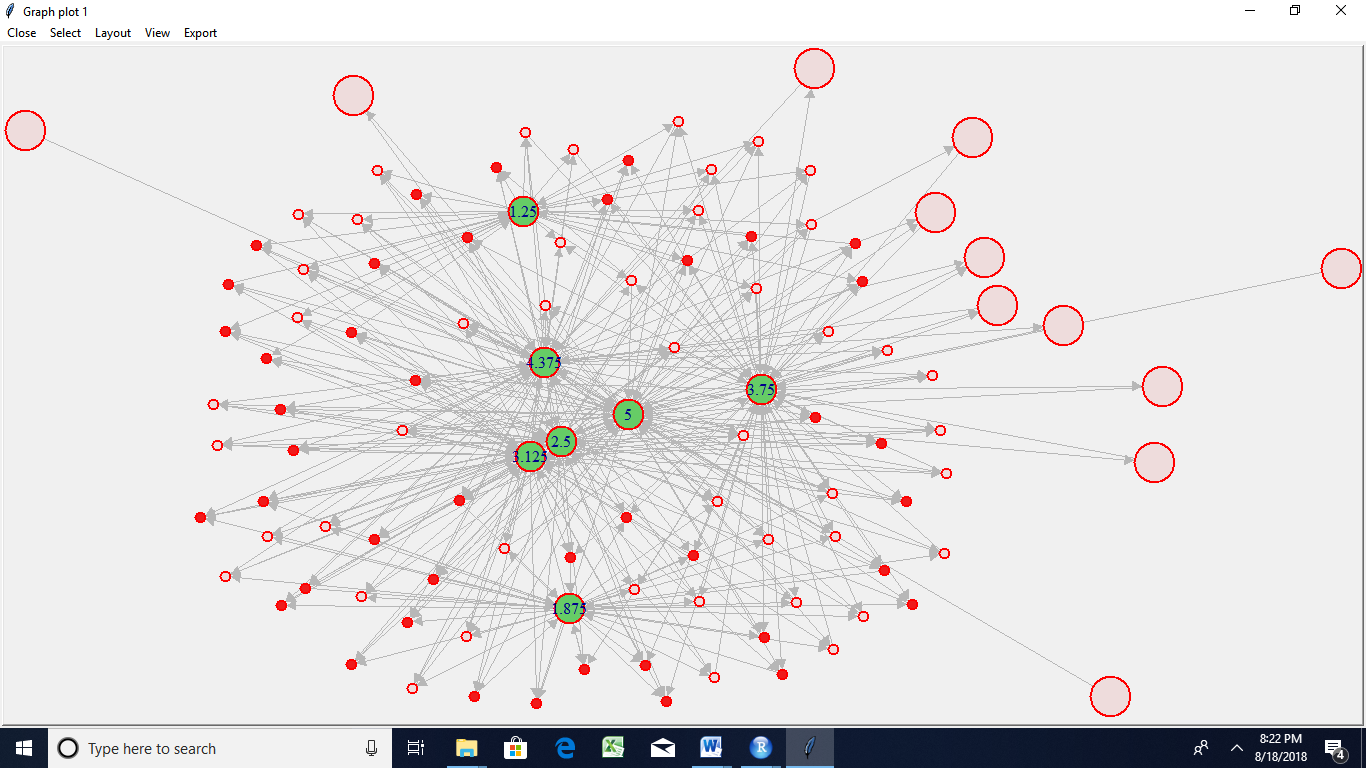
## Dim.1 Dim.2 Dim.3 Dim.4 Dim.5  
## rating -0.19 0.945 0.267 0.0018 -0.0084  
## calories -0.95 -0.105 0.178 -0.1581 0.1674  
## protein -0.87 -0.085 0.045 0.4911 -0.0223  
## fat -0.92 -0.169 0.207 -0.2514 -0.1473  
## sodium -0.59 0.254 -0.765 -0.0752 -0.0051

res.var$contrib # Contributions to the PCs

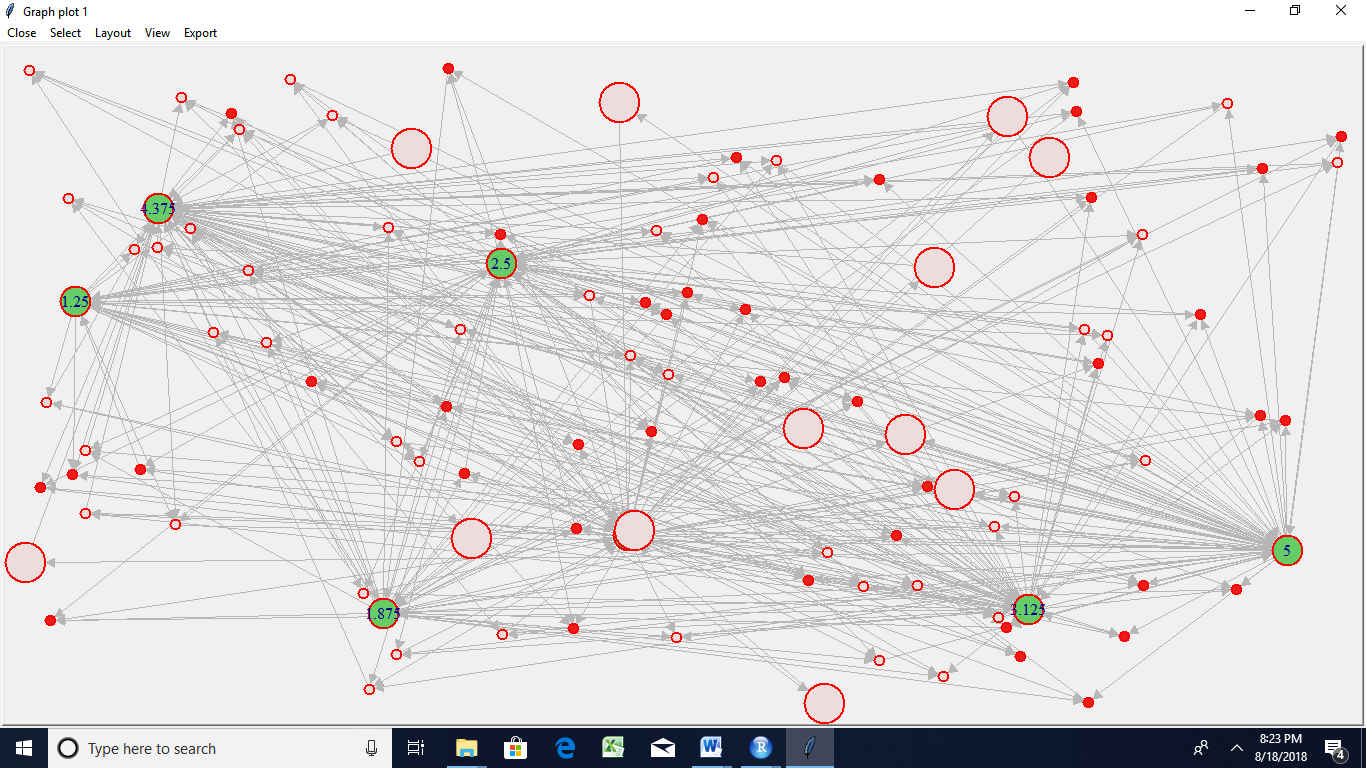
## Dim.1 Dim.2 Dim.3 Dim.4 Dim.5  
## rating 1.3 88.89 9.70 9.9e-04 0.141  
## calories 31.4 1.10 4.32 7.5e+00 55.692  
## protein 26.0 0.72 0.28 7.2e+01 0.986  
## fat 29.3 2.86 5.82 1.9e+01 43.130  
## sodium 12.0 6.42 79.89 1.7e+00 0.052

res.var$cos2 # Quality of representation

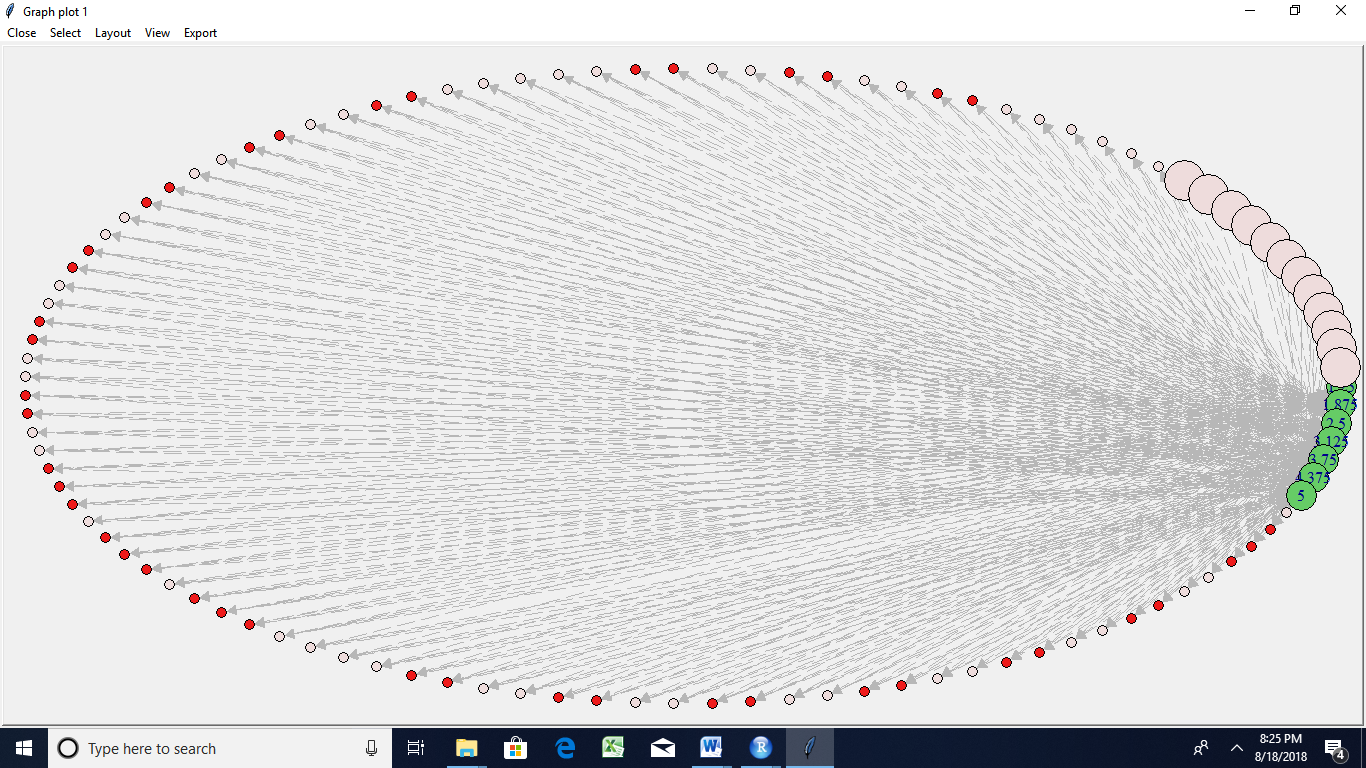
## Dim.1 Dim.2 Dim.3 Dim.4 Dim.5  
## rating 0.037 0.8922 0.071 3.3e-06 7.1e-05  
## calories 0.904 0.0111 0.032 2.5e-02 2.8e-02  
## protein 0.749 0.0073 0.002 2.4e-01 5.0e-04  
## fat 0.844 0.0287 0.043 6.3e-02 2.2e-02  
## sodium 0.344 0.0645 0.586 5.7e-03 2.6e-05

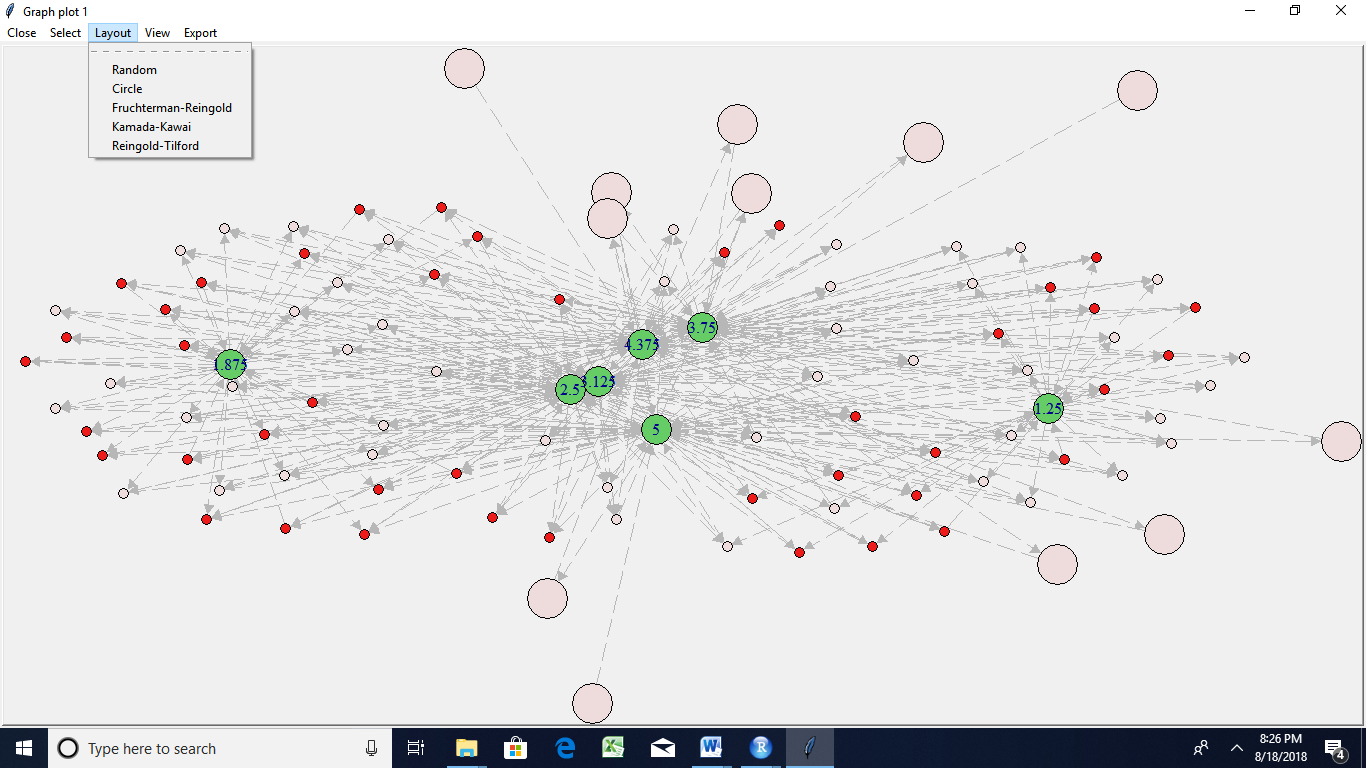


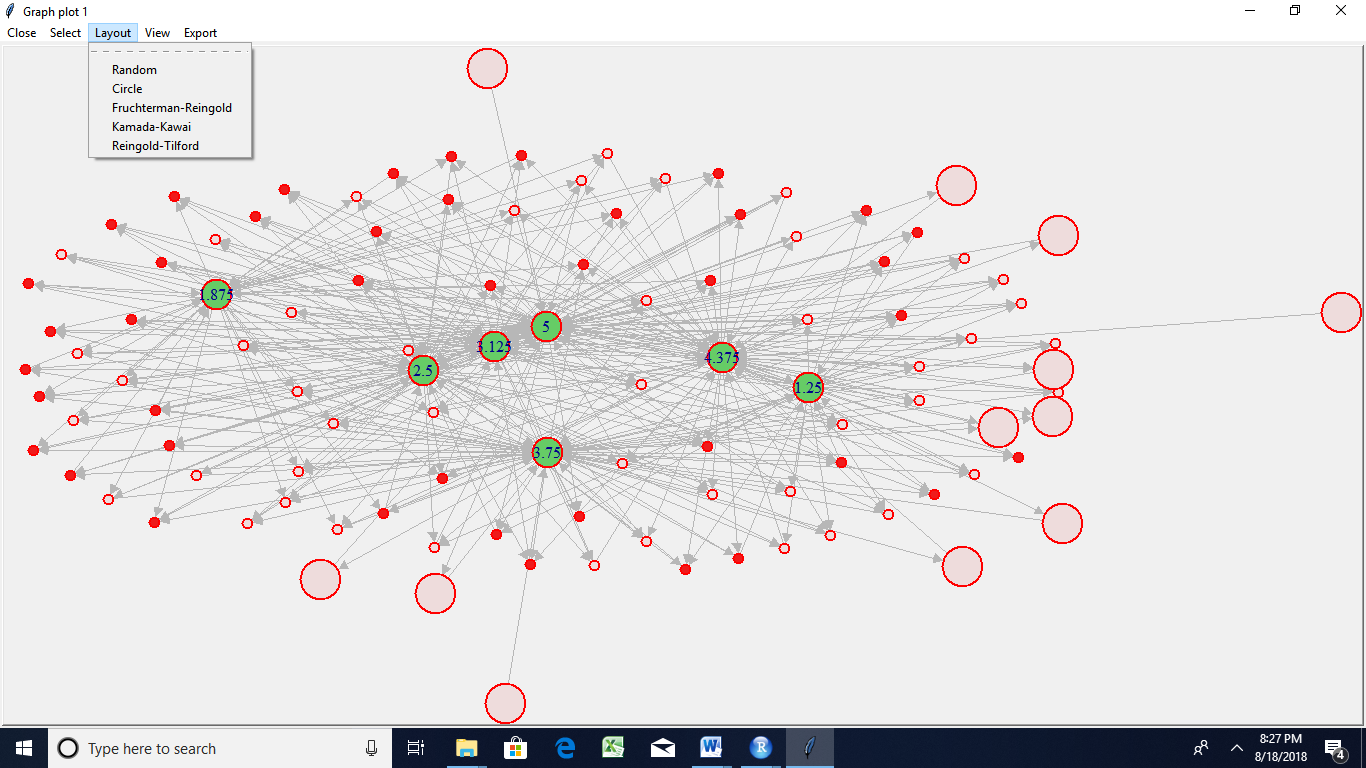
Random type



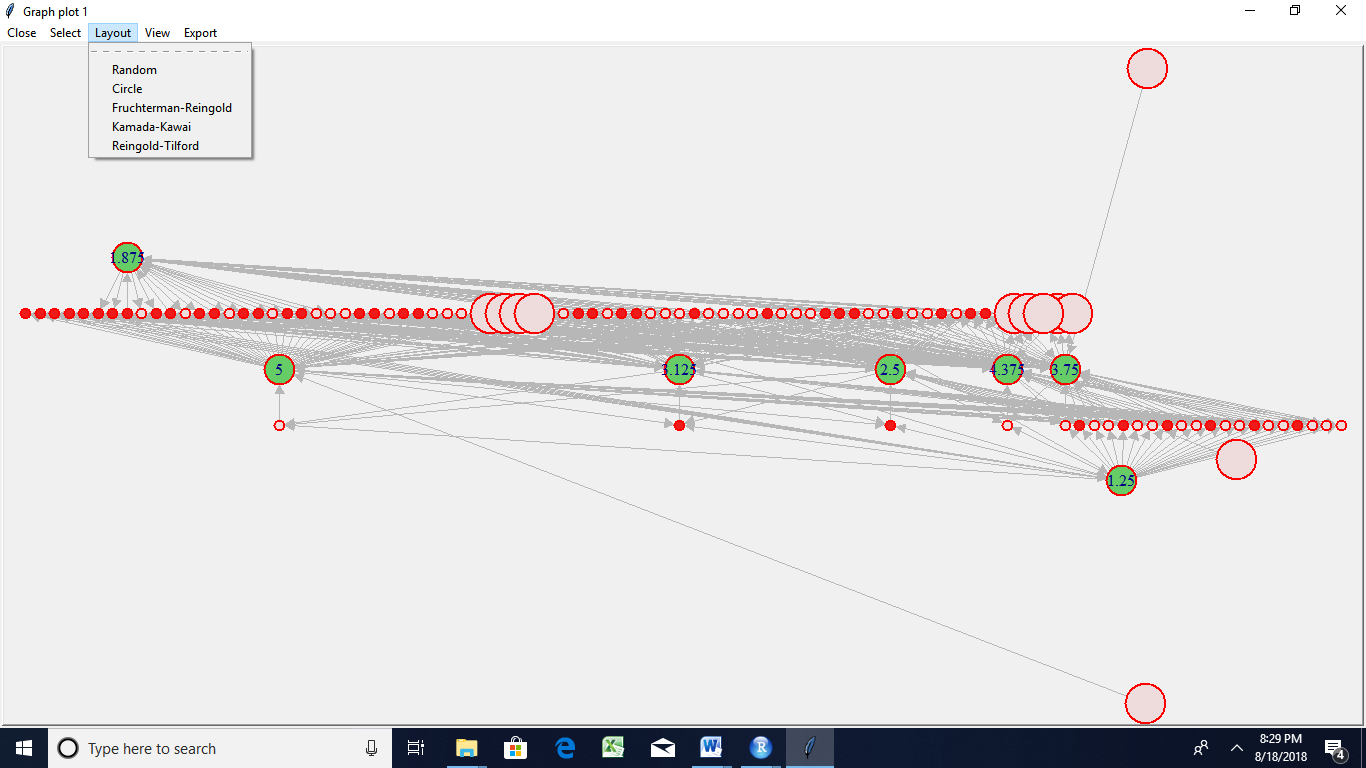
Circle







Reingold



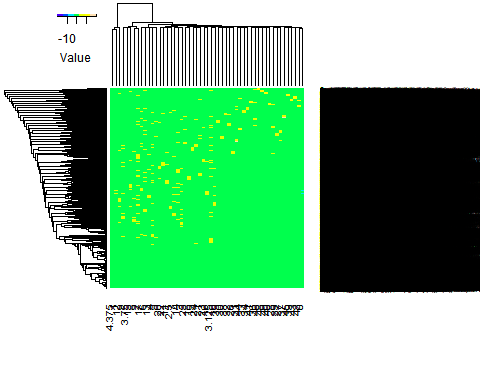
Additional trials

library(readr)  
epi\_r<- read.csv("C:/Users/Jagannath/Documents/assignment data acadgild/assignmnet 21-24/epi\_r.csv")

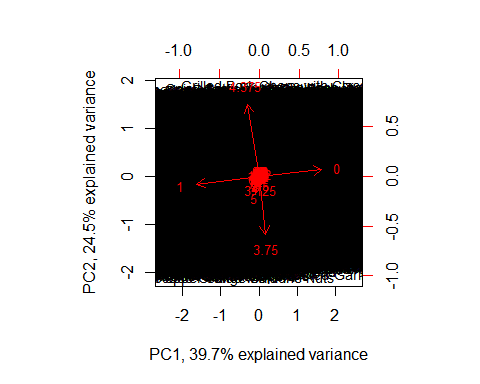
View(epi\_r)  
data<-epi\_r  
View(data)  
  
a <- aggregate(data[,-1], by=list(data[,1]), paste, collapse=",")  
a$combined <- apply(a[,2:ncol(a)], 1, paste, collapse=",")  
a$combined <- gsub(",NA","",a$combined) ## this column contains the totality of all ingredients for a cuisine  
  
cuisines <- as.data.frame(table(data[,1])) ## Number of recipes for each cuisine  
freq <- lapply(lapply(strsplit(a$combined,","), table), as.data.frame) ## Frequency of ingredients  
names(freq) <- a[,1]  
prop <- lapply(seq\_along(freq), function(i) { colnames(freq[[i]])[2] <- names(freq)[i] freq[[i]][,2] <- freq[[i]][,2]/cuisines[i,2] ## proportion (normalized frequency) freq[[i]]})  
names(prop) <- a[,1] ## this is a list of 26 elements, one for each cuisine  
final <- Reduce(function(...) merge(..., all=TRUE, by="Var1"), prop)  
row.names(final) <- final[,1]  
final <- final[,-1]  
final[is.na(final)] <- 0 ## If ingredient missing in all recipes, proportion set to zero  
final <- t(final) ## proportion matrix  
s <- sort(apply(final, 2, sd), decreasing=TRUE)  
## Selecting ingredients with maximum variation in frequency among cuisines and  
## Using standardized proportions for final analysis  
final\_imp <- scale(subset(final, select=names(which(s > 0.1))))   
## heatmap   
library(gplots) ##   
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':  
##   
## lowess

heatmap.2(final\_imp, trace="none", margins = c(6,11), col=topo.colors(7),   
 key=TRUE, key.title=NA, keysize=1.2, density.info="none")



## PCA and biplot   
p <- princomp(final\_imp)   
biplot(p,pc.biplot=TRUE, col=c("black","red"), cex=c(0.9,0.8),xlim=c(-2.5,2.5), xlab="PC1, 39.7% explained variance", ylab="PC2, 24.5% explained variance")



|  |
| --- |
|  |
| #1. Use the below given data set |
|  | #Data Set |
|  | #2. Perform the below given activities: |
|  | #a. Apply PCA to the dataset and show proportion of variance |
|  | #b. Perform PCA using SVD approach |
|  | #c. Show the graphs of PCA components |
|  |  |
|  |  |
|  | #Answers |
|  | #a),b),c) |
|  | #reading the dataset |
|  | #using iris dataset |
|  |  |
|  | data.iris= read.csv  # read iris dataset |
|  |  |
|  | View(data.iris) |
|  |  |
|  | str(data.iris) |
|  |  |
|  | # Checking the data set with boxplot |
|  | boxplot(data.iris[,-5]) |
|  | plot(SepalLengthCm~SepalWidthCm,data.iris) |
|  |  |
|  | # Normalization |
|  | # data.iris[,-5] , here -5 is just to remove species variable which is a factor. |
|  |  |
|  | m<-apply(data.iris[,-5],2,mean) #generating mean for all the variables |
|  |  |
|  | sd<-apply(data.iris[,-5],2,sd) # generating standard deviation for all the variables |
|  |  |
|  | z<-scale(data.iris[,-5],m,sd) # Scaling it |
|  |  |
|  | boxplot(z,horizontal=T) # boxplot to check data variability within variables after normilization |
|  |  |
|  |  |
|  | #PCA USING SPECTRAL DECOMPOSITION IN R |
|  | pc<- PC(data.iris[,-5],method="eigen",scaled=T,graph=F,rm.na=T,print.results=T) |
|  |  |
|  | pc1.cor<-princomp(data.iris[,-5], cor=TRUE) #PCA performed with correlation matrix |
|  | pc1.cor |
|  |  |
|  | #PCA USING SINGULAR VALUE DECOMPOSITION IN R method |
|  | library(factoextra) |
|  |  |
|  | pca1<- prcomp(data.iris[,-5],scale=T) |
|  | pca1 |
|  |  |
|  | #this command show most useful info |
|  | summary(pca1) |
|  |  |
|  | #outputs the mean of variables |
|  | pca1$center |
|  |  |
|  | #scale |
|  | pca1$scale |
|  |  |
|  | #The rotation measure provides the principal component loading |
|  | pca1$rotation |
|  |  |
|  | #standard deviation |
|  | pca1$sdev |
|  |  |
|  | #the principal component score vector |
|  | pca1$x |
|  |  |
|  | #Proportion of Variance |
|  | summary(pca1)$importance[2,] |
|  |  |
|  | #Cumulative Proportion |
|  | summary(pca1)$importance[3,] |
|  |  |
|  | #The variance explained by each principal component |
|  | VE <- pca1$sdev^2 |
|  | VE |
|  |  |
|  | #the proportion of variance explained by each principal component |
|  | PVE <- VE / sum(VE) |
|  | PVE |
|  |  |
|  | #c) |
|  | #answers |
|  | #graphs |
|  | #scree plot |
|  | fviz\_eig(pca1) |
|  | fviz\_screeplot(pca1) |
|  |  |
|  | #more |
|  | plot(pca1, type = "l") |
|  |  |
|  | #plot the resultant principal components |
|  | biplot(pca1, scale = 0) |
|  |  |
|  | #ggbiplot |
|  | library(ggbiplot) |
|  | g <- ggbiplot(pca1, obs.scale = 1, var.scale = 1, |
|  | groups = ir.species, ellipse = TRUE, |
|  | circle = TRUE) |
|  | g <- g + scale\_color\_discrete(name = '') |
|  | g <- g + theme(legend.direction = 'horizontal', |
|  | legend.position = 'top') |
|  | print(g) |
|  |  |
|  | #individuals pca |
|  | fviz\_pca\_ind(pca1, |
|  | col.ind = "cos2", # Color by the quality of representation |
|  | gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"), |
|  | repel = TRUE # Avoid text overlapping |
|  | ) |
|  |  |
|  | #variables pca |
|  | fviz\_pca\_var(pca1, |
|  | col.var = "contrib", # Color by contributions to the PC |
|  | gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"), |
|  | repel = TRUE # Avoid text overlapping |
|  | ) |
|  |  |
|  | #biplot pca |
|  | fviz\_pca\_biplot(pca1, repel = TRUE, |
|  | col.var = "#2E9FDF", # Variables color |
|  | col.ind = "#696969" # Individuals color |
|  | ) |
|  |  |
|  |  |