Market Basket Analysis

Andreu Oros, Sergi Pallice, Joël Ribera

2019-07-24

Contents

Executive Summary

Introduction

Danielle has asked the team to perform a market basket analysis to help Blackwell's board of directors better understand the clientele that Electronidex is currently serving and if Electronidex would be an optimal acquisition.

A dataset of transactions has been provided. The dataset contains 9835 transactions and 125 different products over a 30-day period, or about 327 transactions a day. This tells us the retailer is neither large, nor small.

Results and conclussions

After the analysis, we conclude Electronidex's sales are to be categorized in two forms: retail (B2C) and corporate (B2B). Had we had this information previous to our analysis, it would have saved time in the exploration phase.

Interesting patterns and item relationships found: Retail: >

Would Blackwell benefit in selling Electronidex's items?

Limitations and observations:

Properties of the dataset: 1. The iMac is the product most bought, in 20% of all transactions. This high number stands out considering the large variety of products, especially being the iMac a pricey product. If this number is representative of all sales throughout the year, then Electronidex is potentially profitable. 2. The mean of items bought per transaction is almost 5. Logically, I would say most people in the real world would buy 1 or 2 items per transactions most frequently, in an eletronics store.

Recommendations

Blackwell should acquire Electronidex If Blackwell does acquire Electronidex, do you have any recommendations for Blackwell? (Ex: cross-selling #items, sale promotions, should they remove items, etc.)

Loading Packages and importing datasets

```
pacman::p_load(readr, rstudioapi, ggplot2, party, dplyr, arules, arulesViz,
    RColorBrewer, readxl, tidyr, Hmisc)
setwd("..")
trans <- read.transactions("./Datasets/ElectronidexTransactions2017.csv", format = "basket",
    sep = ",", rm.duplicates = TRUE)
labels <- read_excel("./Datasets/ElectronidexItems2017.xlsx")
```

Preprocessing

```
labels[is.na(labels)] <- "Unknown"
summary(trans)
```

```
transactions as itemMatrix in sparse format with
 9835 rows (elements/itemsets/transactions) and
 125 columns (items) and a density of 0.03506172

most frequent items:
                   iMac               HP Laptop CYBERPOWER Gamer Desktop
                   2519                    1909                    1809
          Apple Earpods        Apple MacBook Air                  (Other)
                   1715                    1530                   33622

element (itemset/transaction) length distribution:
sizes
   0    1    2    3    4    5    6    7    8    9   10   11   12   13   14
   2 2163 1647 1294 1021  856  646  540  439  353  247  171  119   77   72
  15   16   17   18   19   20   21   22   23   25   26   27   29   30
  56   41   26   20   10   10   10    5    3    1    1    3    1    1

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   2.000   3.000   4.383   6.000  30.000

includes extended item information - examples:
                           labels
1 1TB Portable External Hard Drive
2 2TB Portable External Hard Drive
3                  3-Button Mouse
```

```
trans@itemInfo$labels <- labels$ProductType
trans@itemInfo$category <- labels$Category

trans_matrix <- as(trans, "matrix")
```

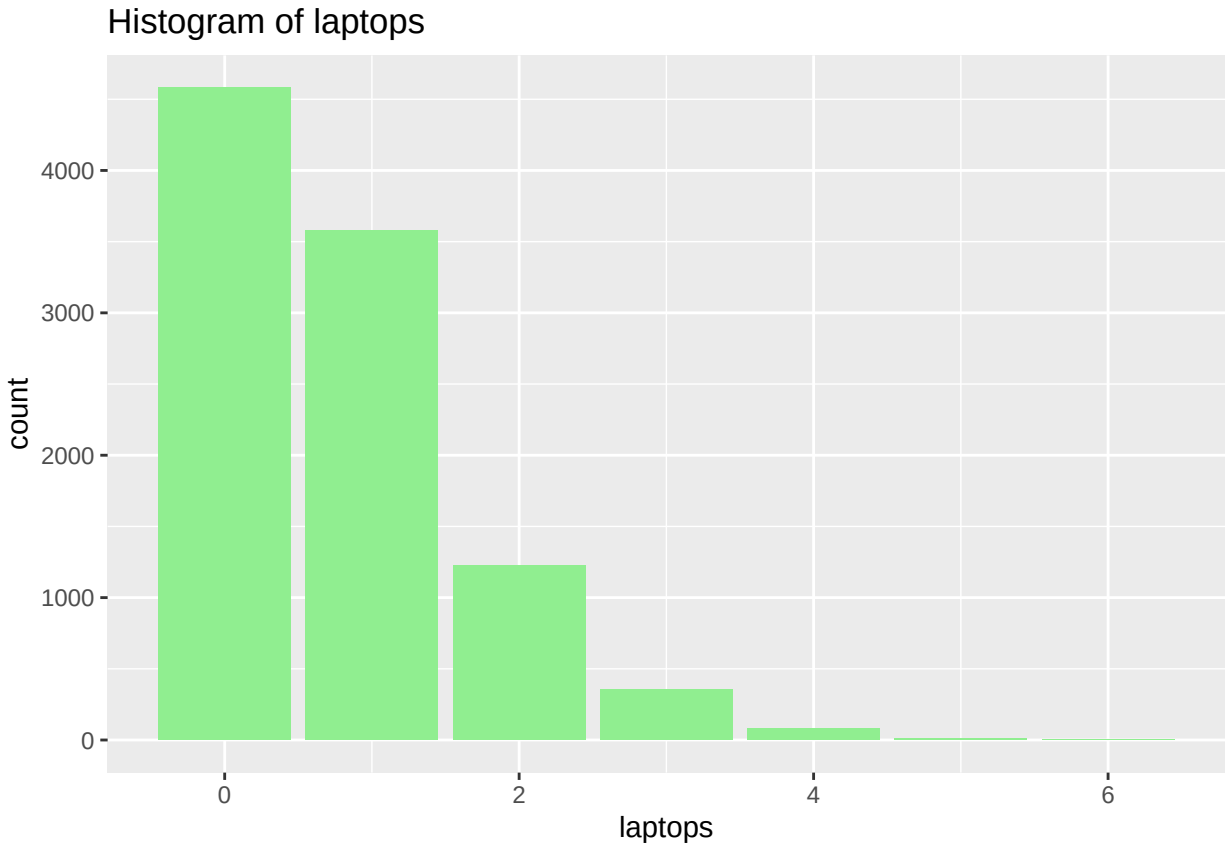
```r
trans_df <- as.data.frame(trans_matrix)
# Turning into a sparematrix of 1s and 0s.
for (i in 1:ncol(trans_df)) {
    trans_df[, i] <- as.integer(trans_df[, i])
}
```

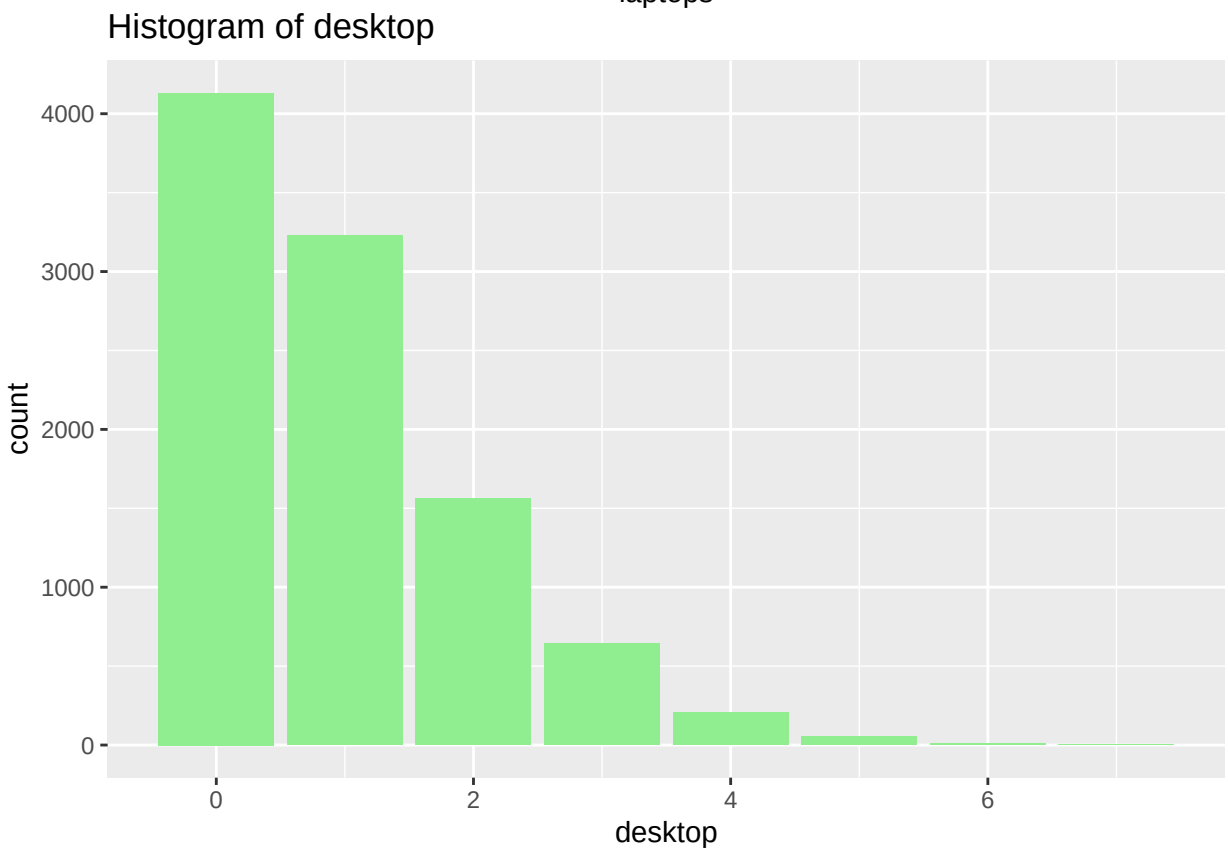
Feature Engineering

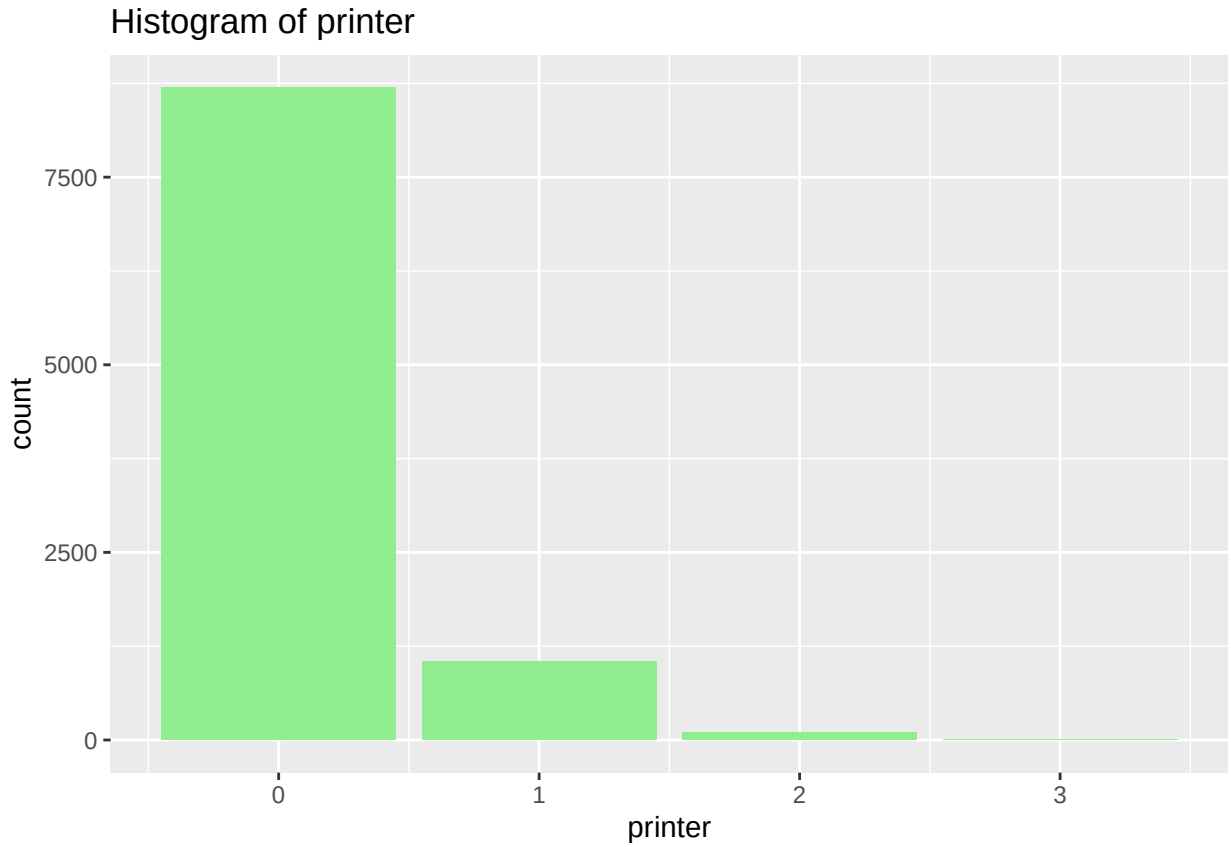
```r
# Counting number of items
nitems <- c()
for (i in 1:nrow(trans_df)) {
    nitems <- c(nitems, sum(trans_df[i, ]))
}
trans_df$nitems <- nitems
trans_df$laptops <- trans_df[, which(colnames(trans_df) == "LG Touchscreen Laptop")] +
    trans_df[, which(colnames(trans_df) == "Acer Aspire")] + trans_df[, which(colnames(trans_df) ==
    "HP Laptop")] + trans_df[, which(colnames(trans_df) == "ASUS Chromebook")] +
    trans_df[, which(colnames(trans_df) == "Apple Macbook Pro")] + trans_df[,
    which(colnames(trans_df) == "Apple MacBook Air")] + trans_df[, which(colnames(trans_df) ==
    "Dell Laptop")] + trans_df[, which(colnames(trans_df) == "Eluktronics Pro Gaming Laptop")] +
    trans_df[, which(colnames(trans_df) == "Alienware AW17R4-7345SLV-PUS 17\" Laptop")] +
    trans_df[, which(colnames(trans_df) == "HP Notebook Touchscreen Laptop PC")]
trans_df$desktop <- trans_df[, which(colnames(trans_df) == "Lenovo Desktop Computer")] +
    trans_df[, which(colnames(trans_df) == "iMac")] + trans_df[, which(colnames(trans_df) ==
    "HP Desktop")] + trans_df[, which(colnames(trans_df) == "ASUS Desktop")] +
    trans_df[, which(colnames(trans_df) == "Dell Desktop")] + trans_df[, which(colnames(trans_df) ==
    "Intel Desktop")] + trans_df[, which(colnames(trans_df) == "Acer Desktop")] +
    trans_df[, which(colnames(trans_df) == "CYBERPOWER Gamer Desktop")] + trans_df[,
    which(colnames(trans_df) == "Dell 2 Desktop")]
trans_df$tablet <- trans_df[, which(colnames(trans_df) == "iPad")] + trans_df[,
    which(colnames(trans_df) == "iPad Pro")] + trans_df[, which(colnames(trans_df) ==
    "Fire HD Tablet")] + trans_df[, which(colnames(trans_df) == "Samsung Galaxy Tab")] +
    trans_df[, which(colnames(trans_df) == "Kindle")]
trans_df$printer <- trans_df$`Epson Printer` + trans_df$`HP Wireless Printer` +
    trans_df$`Canon Office Printer` + trans_df$`Brother Printer` + trans_df$`DYMO Label Manker`
trans_df$nmain <- trans_df$printer + trans_df$laptops + trans_df$desktop + trans_df$tablet
trans_df$ncomp <- trans_df$nitems - trans_df$nmain
trans_df$value <- 10 * trans_df$nmain + trans_df$ncomp
```

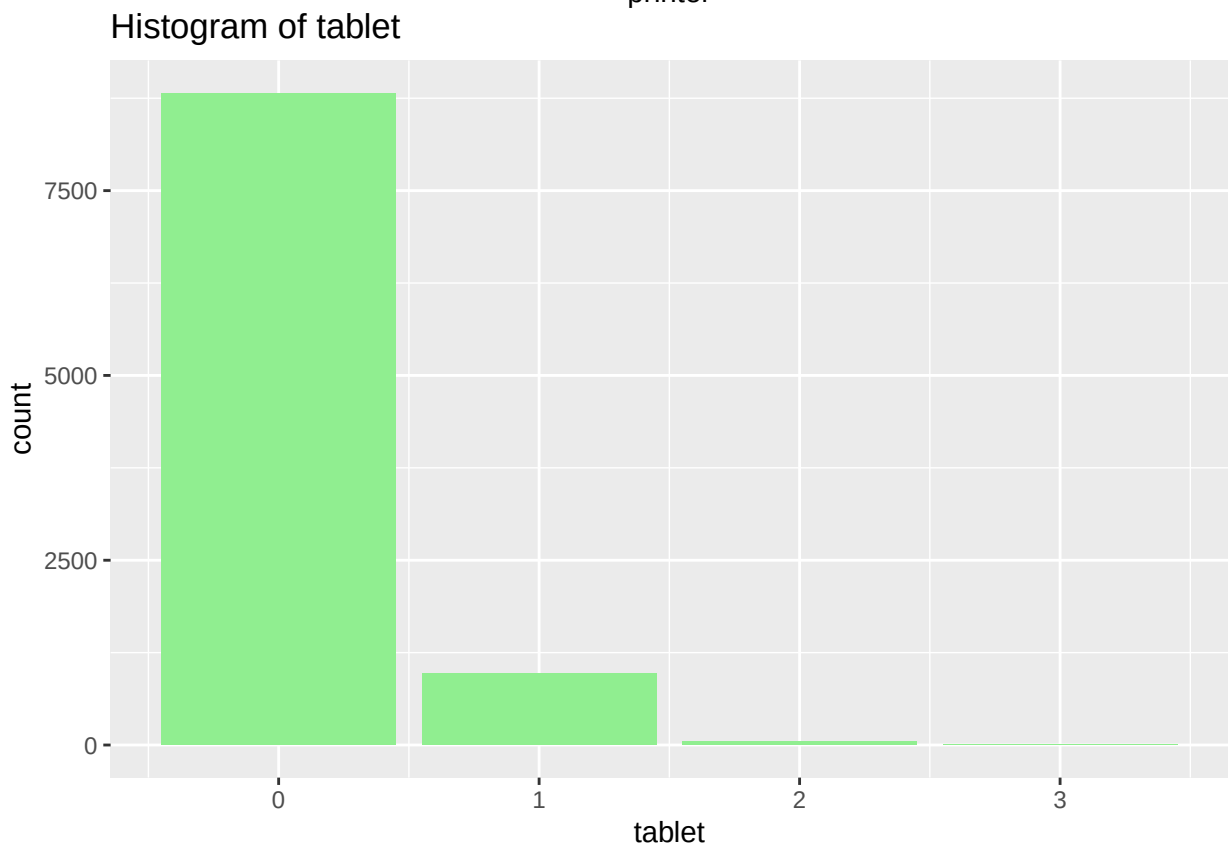
Visualizations

```r
products <- c("laptops", "desktop","printer","tablet")
for (i in products){
  print(ggplot(trans_df, aes_string(x = i)) +
          geom_bar(fill = "lightgreen", bins = 100) +

      ggtitle(paste("Histogram of",i)))
}
```

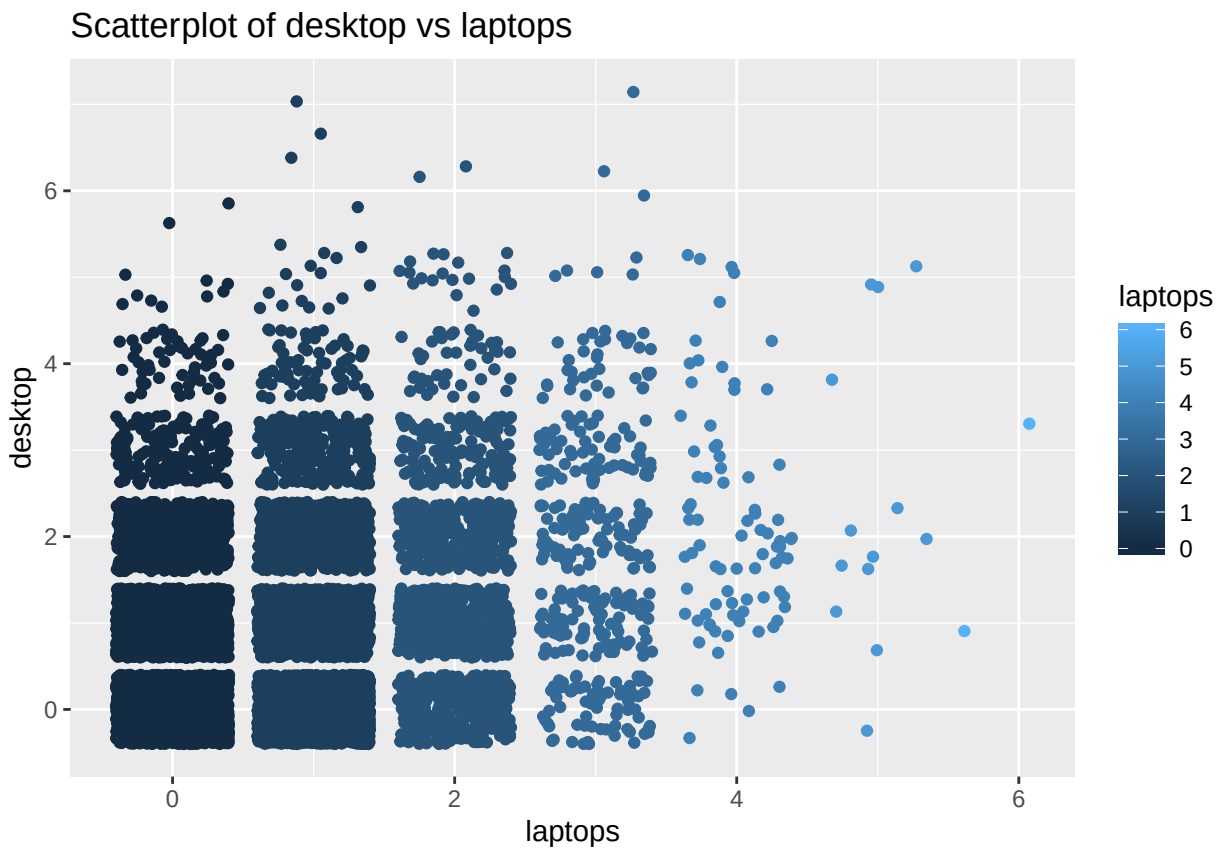

Histogram of laptops
count
4000
3000
2000
1000
0
0
2
4
6
laptops

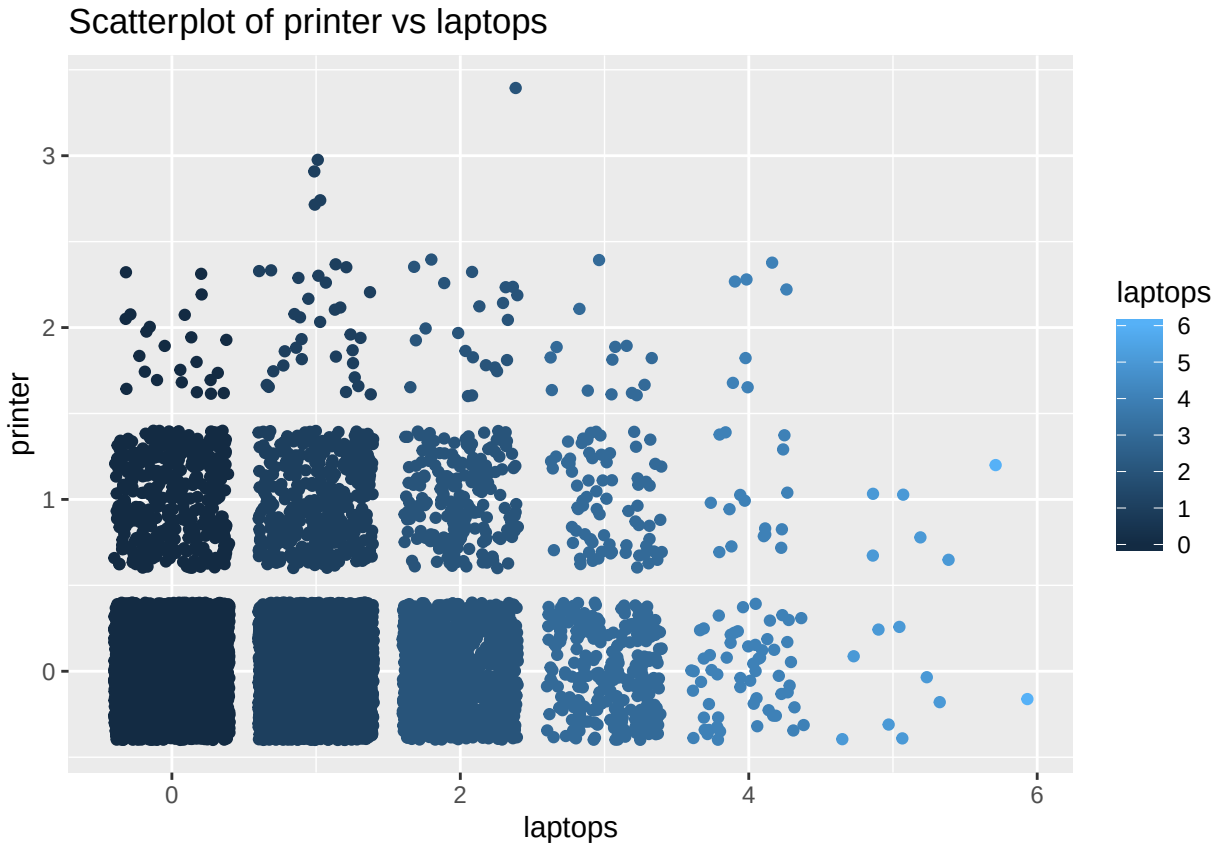

Histogram of desktop
count
4000
3000
2000
1000
0
0
2
4
6
desktop

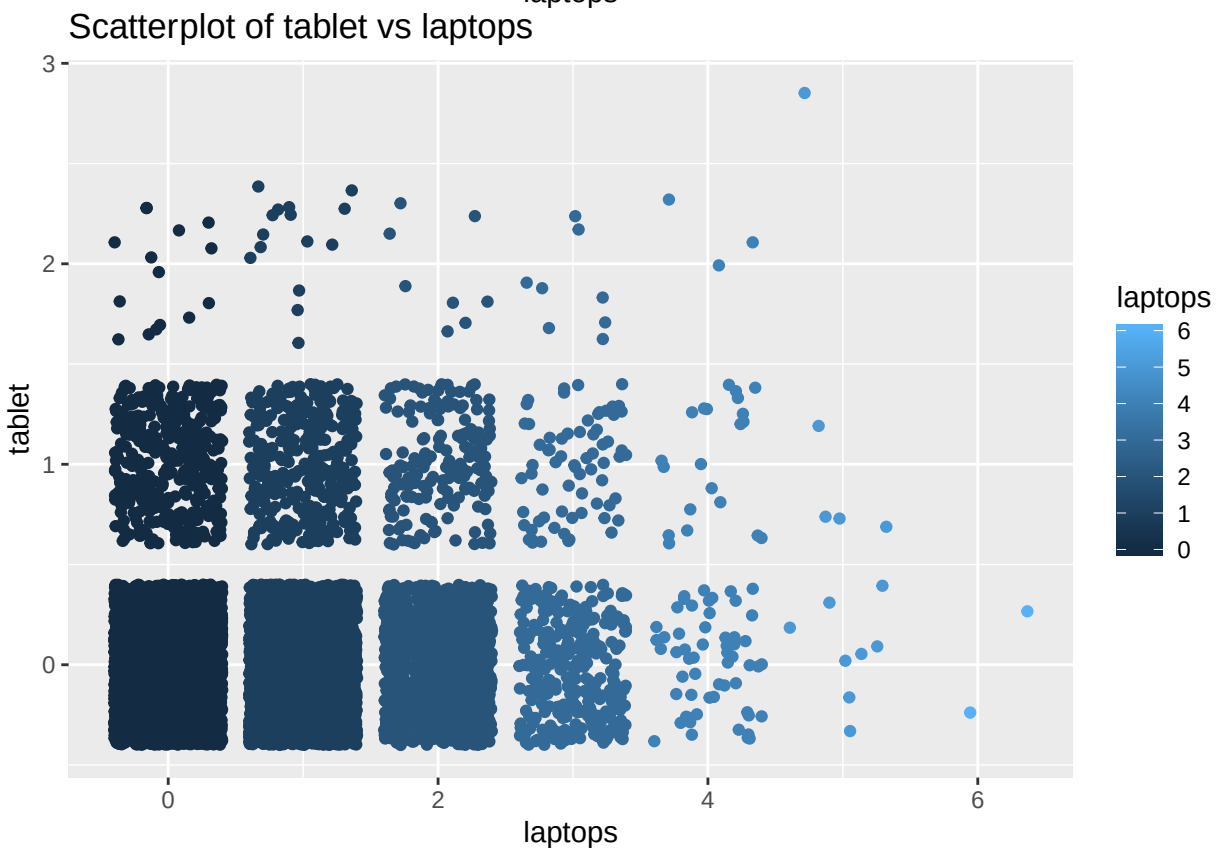

Histogram of printer
count
7500
5000
2500
0
0
1
2
3
printer

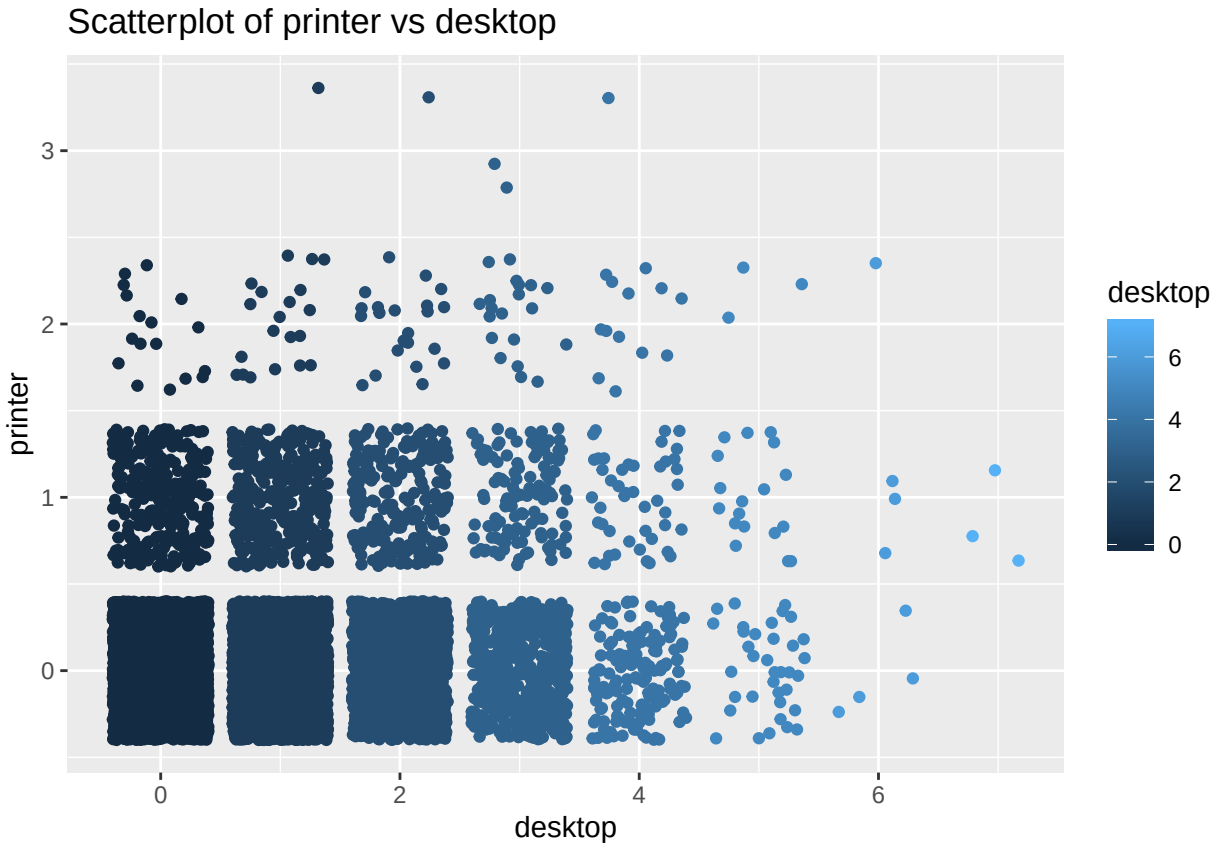

Histogram of tablet
count
7500
5000
2500
0
0
1
2
3
tablet

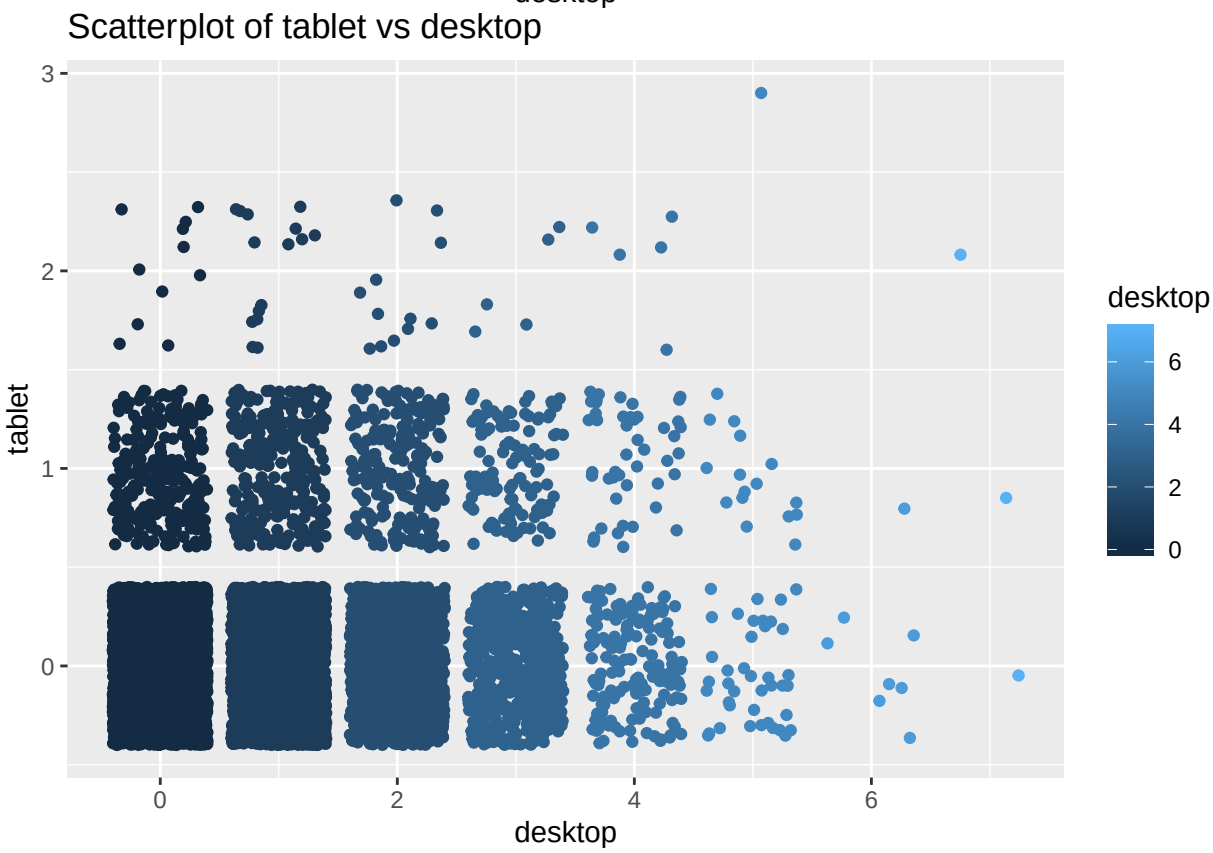
```
for (i in products) {
    for (j in products[-1]) {
        if (i != j | j != i) {
            print(ggplot(trans_df, aes_string(x = i, y = j, color = i)) + geom_jitter() +
                ggtitle(paste("Scatterplot of", j, "vs", i)))

        }

    }
}
```

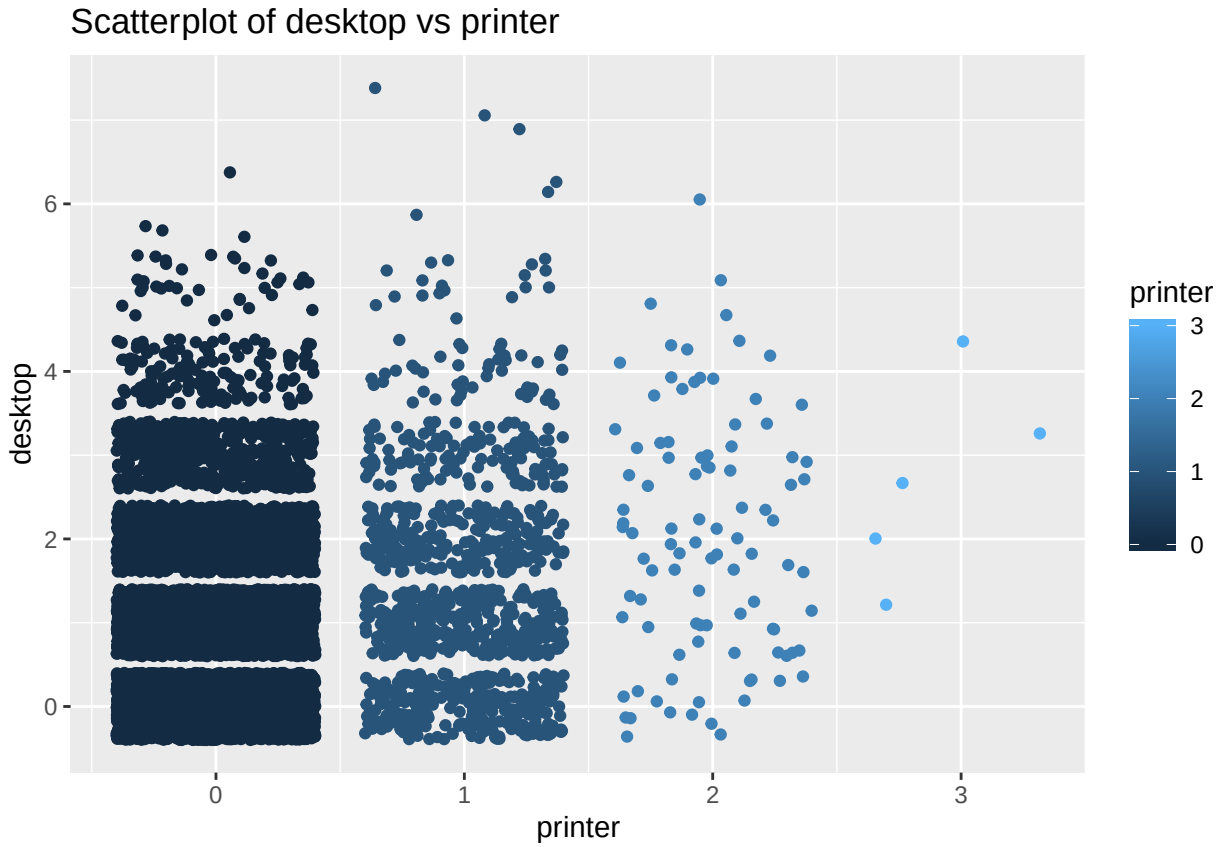


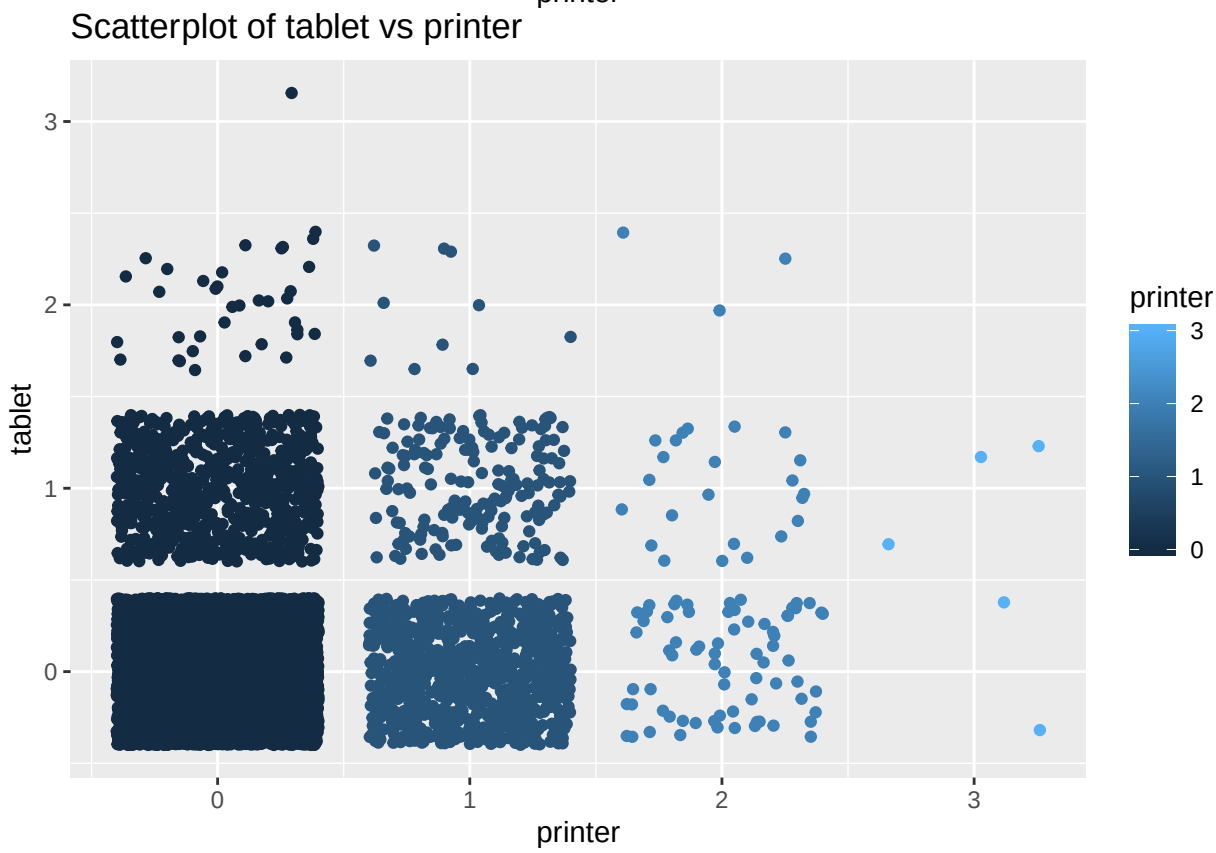

Scatterplot of printer vs laptops
printer
laptops
0
1
2
3
0
2
4
6
laptops
0
1
2
3
4
5
6


Scatterplot of tablet vs laptops
tablet
laptops
0
1
2
3
0
2
4
6
laptops
0
1
2
3
4
5
6

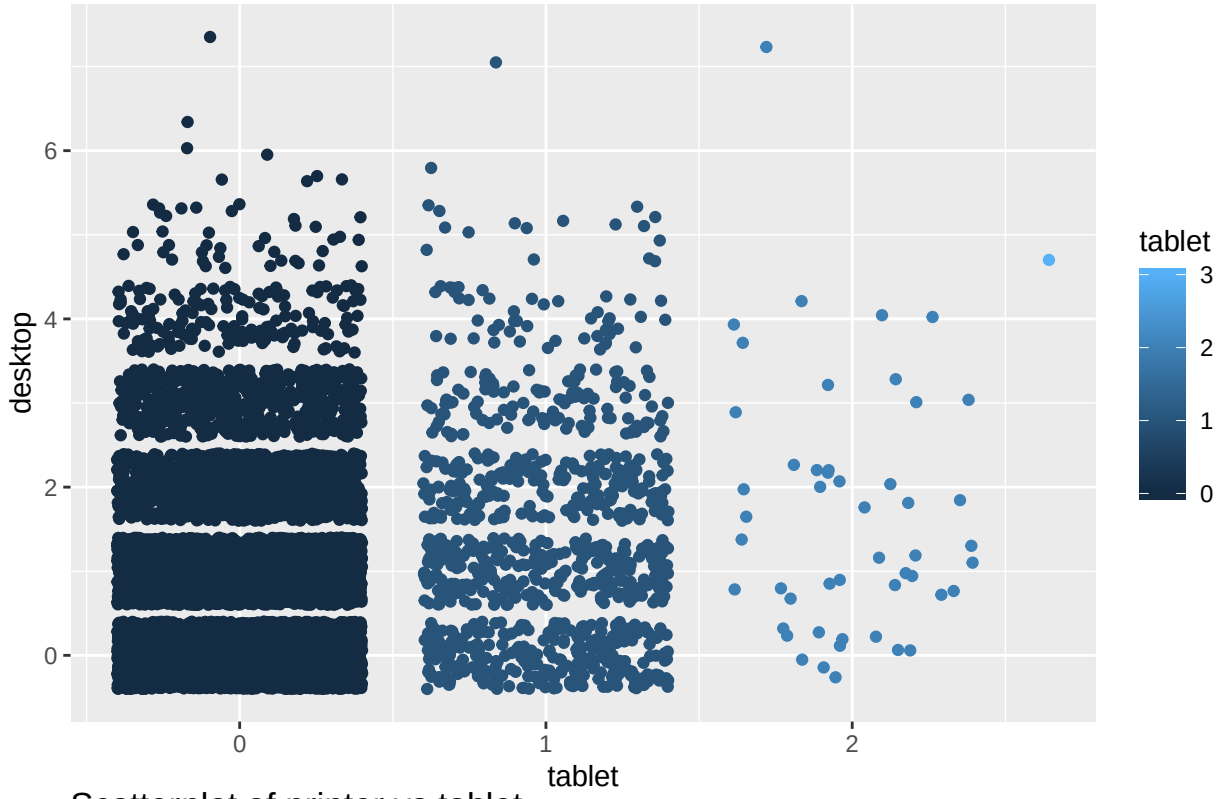

Scatterplot of printer vs desktop
printer
desktop
0
1
2
3
0
2
4
6
desktop
6
4
2
0


Scatterplot of tablet vs desktop
tablet
desktop
0
1
2
3
0
2
4
6
desktop
6
4
2
0

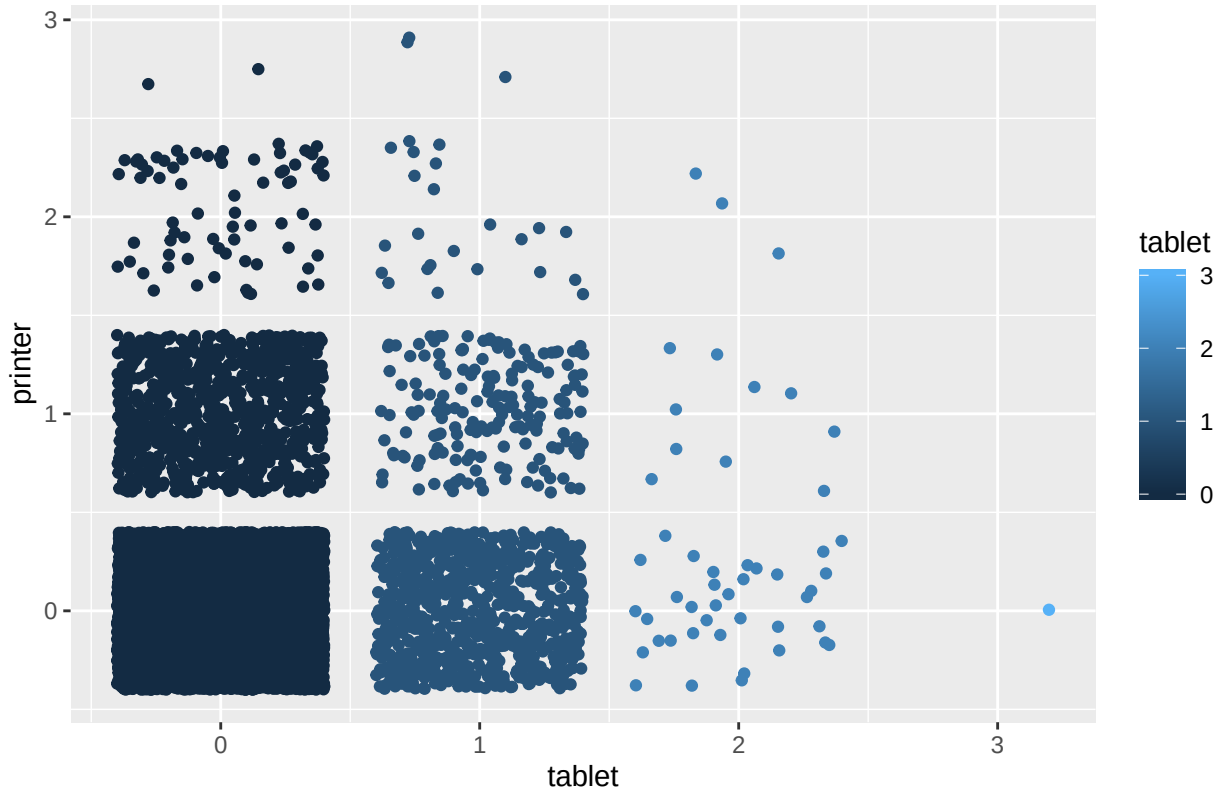

Scatterplot of desktop vs printer
desktop
printer
0
1
2
3
4
6
printer
0
1
2
3


Scatterplot of tablet vs printer
tablet
printer
0
1
2
3
printer
0
1
2
3

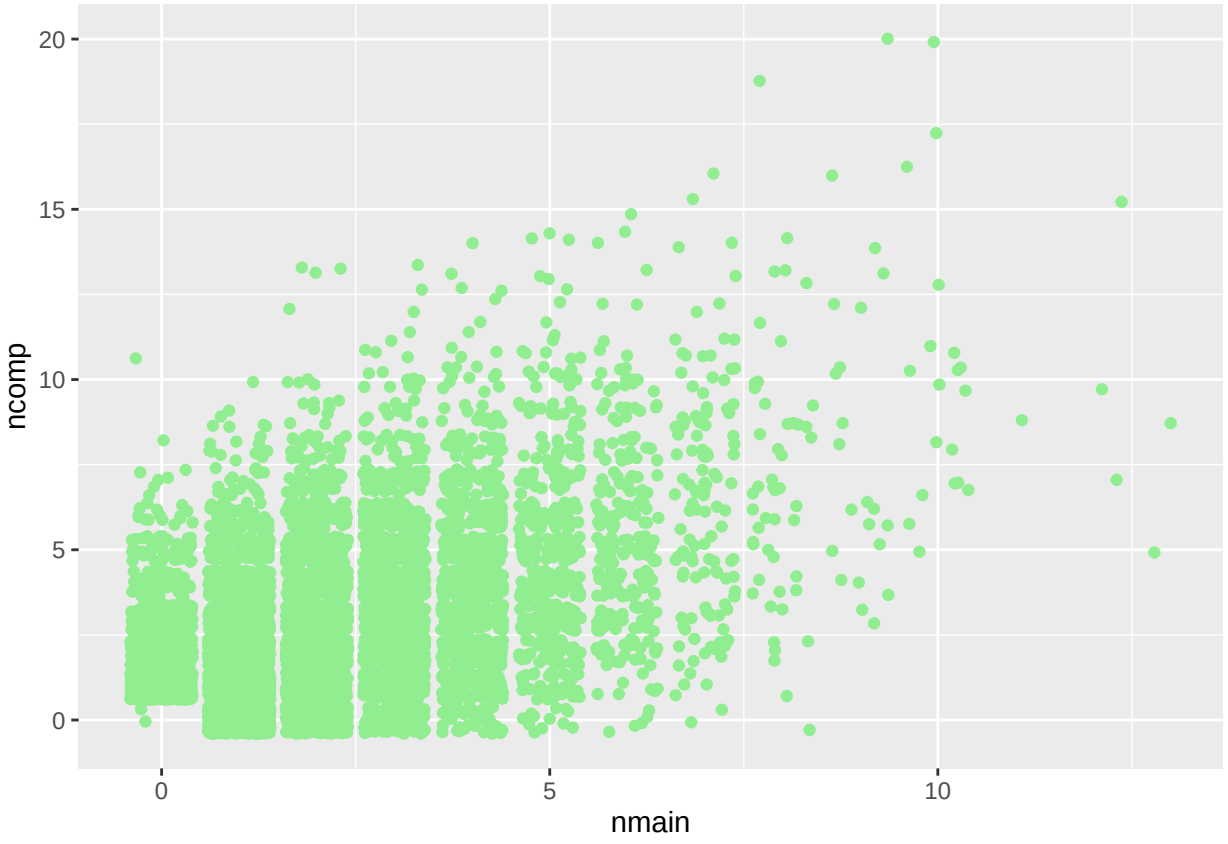
Scatterplot of desktop vs tablet



Scatterplot of printer vs tablet



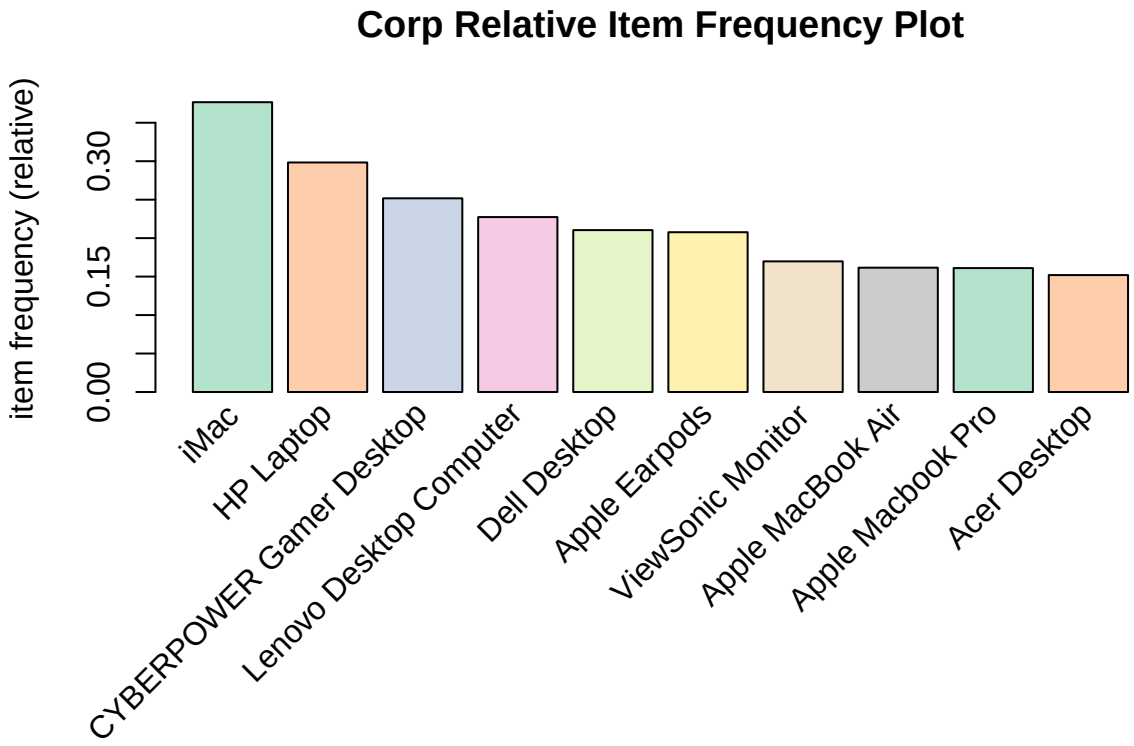
```r
ggplot(trans_df, aes(x = nmain, y = ncomp)) + geom_jitter(color = "lightgreen")
```



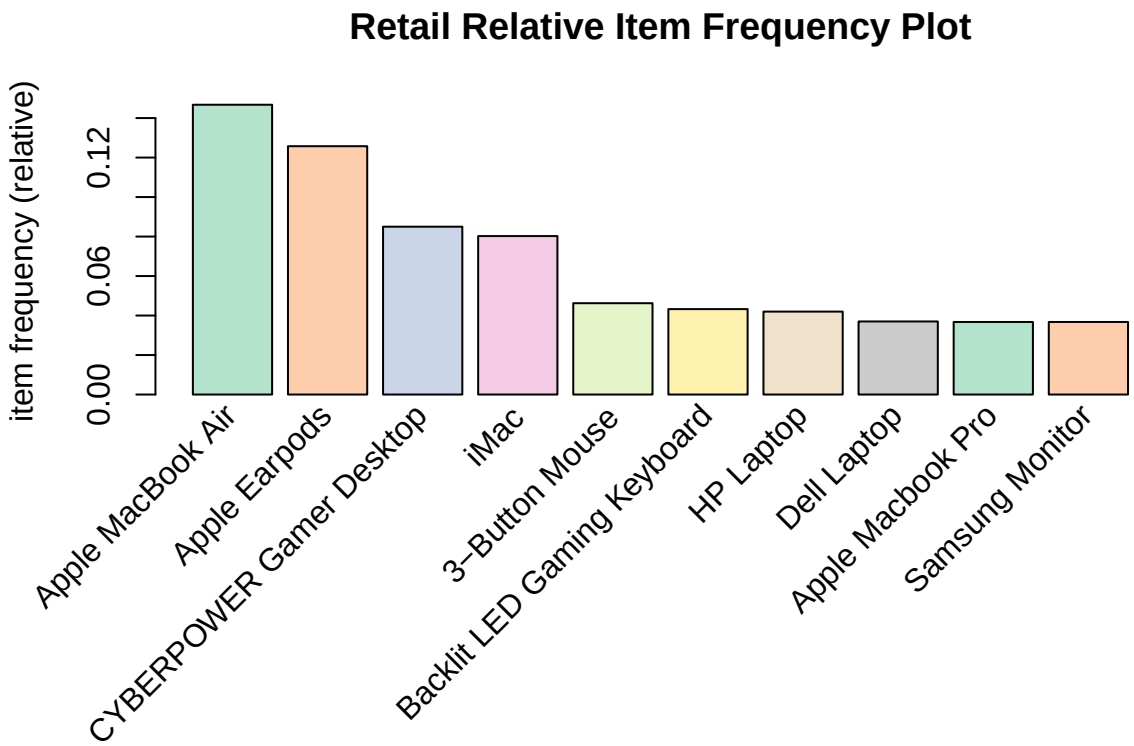
Splitting dataframe between corporates and retailers

```r
# Filtering. Corporate will be those transactions with more than 2 main
# products and 3 complements.
corporate <- filter(trans_df, nmain >= 2 | ncomp >= 3)
# Cleaning the new columns we created before
corporate <- corporate[, -which(colnames(corporate) %in% c("laptops", "desktop",
    "printer", "tablet", "nitems", "nmain", "value", "ncomp"))]
# Filtering. Retailers will be those transactions with less than 2 main
# products and 3 complements.
retailer <- filter(trans_df, nmain <= 1 & ncomp <= 2)
# Cleaning
retailer <- retailer[, -which(colnames(retailer) %in% c("laptops", "desktop",
    "printer", "tablet", "nitems", "nmain", "value", "ncomp"))]
# Transforming the dataframe into a transaction objecte
trans_corp <- as(corporate == 1, "transactions")
trans_retail <- as(retailer == 1, "transactions")
# Inserting labels and the level category
trans_corp@itemInfo$labels <- labels$ProductType
trans_corp@itemInfo$category <- labels$Category
trans_retail@itemInfo$labels <- labels$ProductType
trans_retail@itemInfo$category <- labels$Category
```

```
itemFrequencyPlot(trans_corp, topN = 10, type = "relative", col = brewer.pal(8,
    "Pastel2"), main = "Corp Relative Item Frequency Plot")
```




```
itemFrequencyPlot(trans_retail, topN = 10, type = "relative", col = brewer.pal(8,
    "Pastel2"), main = "Retail Relative Item Frequency Plot")
```



Creating rules via apriori algorithm

Rules for products in corporate transactions

```
rules_corpro <- apriori(trans_corp, parameter = list(supp = 0.01, conf = 0.01,
    minlen = 2, maxlen = 4))
rules_corpro <- rules_corpro[!is.redundant(rules_corpro)]
```

```
summary(rules_corpro)
```

```
set of 1752 rules

rule length distribution (lhs + rhs):sizes
   2    3    4
1032  672   48

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  2.000   2.000   2.000   2.438   3.000   4.000

summary of quality measures:
    support          confidence          lift            count
 Min.   :0.01011   Min.   :0.02775   Min.   :0.6424   Min.   : 59.0
 1st Qu.:0.01148   1st Qu.:0.12435   1st Qu.:1.0782   1st Qu.: 67.0
 Median :0.01422   Median :0.20365   Median :1.2491   Median : 83.0
 Mean   :0.01827   Mean   :0.23648   Mean   :1.3030   Mean   :106.6
 3rd Qu.:0.01971   3rd Qu.:0.32164   3rd Qu.:1.4580   3rd Qu.:115.0
 Max.   :0.12734   Max.   :0.69388   Max.   :2.7337   Max.   :743.0

mining info:
       data ntransactions support confidence
 trans_corp          5835    0.01       0.01
```

```
# The most populars products
corpro_sup <- inspect(head(sort(rules_corpro, by = "support"), 10))
```

```
     lhs                           rhs                          support
[1]  {HP Laptop}                => {iMac}                       0.12733505
[2]  {iMac}                     => {HP Laptop}                  0.12733505
[3]  {Lenovo Desktop Computer}  => {iMac}                       0.09905741
[4]  {iMac}                     => {Lenovo Desktop Computer}    0.09905741
[5]  {CYBERPOWER Gamer Desktop} => {iMac}                       0.09562982
[6]  {iMac}                     => {CYBERPOWER Gamer Desktop}   0.09562982
[7]  {Dell Desktop}             => {iMac}                       0.09203085
[8]  {iMac}                     => {Dell Desktop}               0.09203085
[9]  {ViewSonic Monitor}        => {iMac}                       0.08157669
[10] {iMac}                     => {ViewSonic Monitor}          0.08157669
     confidence lift     count
[1]  0.4267662  1.132930 743
[2]  0.3380346  1.132930 743
[3]  0.4355690  1.156299 578
[4]  0.2629663  1.156299 578
```

```
[5]  0.3798502  1.008383 558
[6]  0.2538672  1.008383 558
[7]  0.4372964  1.160885 537
[8]  0.2443130  1.160885 537
[9]  0.4803229  1.275107 476
[10] 0.2165605  1.275107 476
```

```
# Items that have high chances of being bought together
corpro_con <- inspect(head(sort(rules_corpro, by = "confidence"), 10))
```

```
     lhs                           rhs            support confidence     lift count
[1]  {Dell Desktop,
      Lenovo Desktop Computer,
      ViewSonic Monitor}       => {iMac}      0.01165381  0.6938776 1.842027    68
[2]  {Apple Magic Keyboard,
      ASUS Monitor}            => {iMac}      0.01148243  0.6767677 1.796606    67
[3]  {Acer Aspire,
      iMac,
      ViewSonic Monitor}       => {HP Laptop} 0.01045416  0.6630435 2.222205    61
[4]  {Acer Desktop,
      HP Laptop,
      ViewSonic Monitor}       => {iMac}      0.01079692  0.6562500 1.742138    63
[5]  {ASUS 2 Monitor,
      Dell Desktop}            => {iMac}      0.01525278  0.6449275 1.712080    89
[6]  {ASUS Monitor,
      Lenovo Desktop Computer} => {iMac}      0.01645244  0.6442953 1.710402    96
[7]  {Acer Desktop,
      ASUS 2 Monitor}          => {iMac}      0.01079692  0.6428571 1.706584    63
[8]  {ASUS Monitor,
      ViewSonic Monitor}       => {iMac}      0.01388175  0.6428571 1.706584    81
[9]  {ASUS Monitor,
      Dell Desktop}            => {iMac}      0.01336761  0.6393443 1.697258    78
[10] {Acer Desktop,
      iMac,
      ViewSonic Monitor}       => {HP Laptop} 0.01079692  0.6363636 2.132787    63
```

```
# Lift
corpro_lift <- inspect(head(sort(rules_corpro, by = "lift"), 10))
```

```
     lhs                              rhs
[1]  {HP Black & Tri-color Ink}    => {ViewSonic Monitor}
[2]  {ViewSonic Monitor}           => {HP Black & Tri-color Ink}
[3]  {Acer Aspire,HP Laptop,iMac}  => {ViewSonic Monitor}
[4]  {Dell 2 Desktop}              => {Apple Magic Keyboard}
[5]  {Apple Magic Keyboard}        => {Dell 2 Desktop}
[6]  {Apple Magic Keyboard,iMac}   => {ASUS Monitor}
[7]  {Brother Printer}             => {iPad Pro}
[8]  {iPad Pro}                    => {Brother Printer}
[9]  {HP Laptop,ViewSonic Monitor} => {Computer Game}
[10] {HP Wireless Mouse}           => {Epson Printer}
     support    confidence lift     count
[1]  0.01113967 0.46428571 2.733711 65
[2]  0.01113967 0.06559031 2.733711 65
```

```
[3]  0.01045416 0.46212121 2.720966 61
[4]  0.01439589 0.28378378 2.679415 84
[5]  0.01439589 0.13592233 2.679415 84
[6]  0.01148243 0.21824104 2.551977 67
[7]  0.01028278 0.20547945 2.529478 60
[8]  0.01028278 0.12658228 2.529478 60
[9]  0.01233933 0.15652174 2.509078 72
[10] 0.01165381 0.18630137 2.470610 68
```

Rules for categories in corporate transactions

```
trans_corcat <- aggregate(trans_corp, by = "category")
rules_corcat <- apriori(trans_corcat, parameter = list(supp = 0.01, conf = 0.01,
    minlen = 2, maxlen = 4))
rules_corcat <- rules_corcat[!is.redundant(rules_corcat)]
```

```
summary(rules_corcat)
```

```
set of 3599 rules

rule length distribution (lhs + rhs):sizes
   2    3    4
 262 1191 2146

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  2.000   3.000   4.000   3.523   4.000   4.000

summary of quality measures:
    support          confidence          lift            count
 Min.   :0.01011   Min.   :0.0537   Min.   :0.9325   Min.   :  59.0
 1st Qu.:0.01354   1st Qu.:0.2514   1st Qu.:1.1939   1st Qu.:  79.0
 Median :0.01971   Median :0.3898   Median :1.3394   Median : 115.0
 Mean   :0.03069   Mean   :0.4518   Mean   :1.3618   Mean   : 179.1
 3rd Qu.:0.03325   3rd Qu.:0.7022   3rd Qu.:1.4976   3rd Qu.: 194.0
 Max.   :0.54070   Max.   :0.9474   Max.   :2.4056   Max.   :3155.0

mining info:
         data ntransactions support confidence
 trans_corcat          5835    0.01       0.01
```

```
# The most populars categories
corcat_sup <- inspect(head(sort(rules_corcat, by = "support"), 10))
```

```
     lhs                    rhs        support   confidence lift     count
[1]  {Laptops}           => {Desktop}  0.5407027 0.7879620  1.008059 3155
[2]  {Desktop}           => {Laptops}  0.5407027 0.6917343  1.008059 3155
[3]  {Monitors}          => {Desktop}  0.4327335 0.7868495  1.006636 2525
[4]  {Desktop}           => {Monitors} 0.4327335 0.5536067  1.006636 2525
[5]  {Monitors}          => {Laptops}  0.3797772 0.6905578  1.006345 2216
[6]  {Laptops}           => {Monitors} 0.3797772 0.5534466  1.006345 2216
[7]  {Laptops,Monitors} => {Desktop}  0.3076264 0.8100181  1.036276 1795
```

```
[8]  {Desktop,Monitors} => {Laptops}  0.3076264 0.7108911  1.035976 1795
[9]  {Desktop,Laptops}  => {Monitors} 0.3076264 0.5689382  1.034514 1795
[10] {Computer Mice}    => {Desktop}  0.2904884 0.7821874  1.000672 1695
```

```
# Items that have high chances of being bought together
corcat_con <- inspect(head(sort(rules_corcat, by = "confidence"), 10))
```

```
     lhs                           rhs          support confidence     lift count
[1]  {Accessories,
      Computer Mice,
      Smart Home Devices}       => {Desktop} 0.01233933  0.9473684 1.211992    72
[2]  {Accessories,
      Computer Mice,
      Computer Stands}          => {Desktop} 0.01199657  0.9333333 1.194036    70
[3]  {Computer Mice,
      Keyboard,
      Printer Ink}              => {Desktop} 0.01782348  0.9285714 1.187944   104
[4]  {Accessories,
      Monitors,
      Smart Home Devices}       => {Desktop} 0.01542416  0.9278351 1.187002    90
[5]  {Computer Mice,
      Mouse and Keyboard Combo,
      Printer Ink}              => {Desktop} 0.01216795  0.9220779 1.179637    71
[6]  {Computer Tablets,
      Mouse and Keyboard Combo,
      Printers}                 => {Desktop} 0.01216795  0.9220779 1.179637    71
[7]  {External Hardrives,
      Laptops,
      Monitors}                 => {Desktop} 0.02022279  0.9218750 1.179377   118
[8]  {Accessories,
      Computer Stands,
      Laptops}                  => {Desktop} 0.01696658  0.9166667 1.172714    99
[9]  {Accessories,
      Keyboard,
      Printer Ink}              => {Desktop} 0.01319623  0.9166667 1.172714    77
[10] {External Hardrives,
      Keyboard,
      Laptops}                  => {Desktop} 0.01216795  0.9102564 1.164514    71
```

```
# Lift
corcat_lift <- inspect(head(sort(rules_corcat, by = "lift"), 10))
```

```
     lhs                       rhs                 support confidence     lift count
[1]  {Keyboard,
      Monitors,
      Printers}             => {Printer Ink}     0.01079692  0.2135593 2.405634    63
[2]  {Computer Tablets,
      Keyboard,
      Monitors}             => {Computer Cords} 0.01302485  0.3247863 2.348362    76
[3]  {Desktop,
      Keyboard,
      Printers}             => {Printer Ink}     0.01182519  0.1988473 2.239911    69
[4]  {Computer Tablets,
```

```
      Keyboard,
      Laptops}             => {Computer Cords} 0.01302485  0.3064516 2.215793    76
[5]  {Accessories,
      Computer Tablets,
      Monitors}            => {Printers}       0.01113967  0.3915663 2.169790    65
[6]  {Accessories,
      Monitors,
      Printers}            => {Computer Cords} 0.01096829  0.2976744 2.152330    64
[7]  {Accessories,
      Computer Tablets}    => {Computer Cords} 0.01182519  0.2936170 2.122993    69
[8]  {Keyboard,
      Laptops,
      Printers}            => {Printer Ink}    0.01011140  0.1867089 2.103178    59
[9]  {Computer Mice,
      Computer Tablets,
      Monitors}            => {Computer Cords} 0.01268209  0.2901961 2.098258    74
[10] {Computer Headphones,
      Keyboard,
      Monitors}            => {Computer Cords} 0.01113967  0.2863436 2.070403    65
```

Rules for products in retailers transactions

```
rules_retpro <- apriori(trans_retail, parameter = list(supp = 0.01, conf = 0.01,
    minlen = 2, maxlen = 4))
rules_retpro <- rules_retpro[!is.redundant(rules_retpro)]
```

```
summary(rules_retpro)
```

```
set of 2 rules

rule length distribution (lhs + rhs):sizes
2
2

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
      2       2       2       2       2       2

summary of quality measures:
    support          confidence          lift           count
 Min.   :0.01425   Min.   :0.1133   Min.   :1.333   Min.   :57
 1st Qu.:0.01425   1st Qu.:0.1269   1st Qu.:1.333   1st Qu.:57
 Median :0.01425   Median :0.1405   Median :1.333   Median :57
 Mean   :0.01425   Mean   :0.1405   Mean   :1.333   Mean   :57
 3rd Qu.:0.01425   3rd Qu.:0.1541   3rd Qu.:1.333   3rd Qu.:57
 Max.   :0.01425   Max.   :0.1676   Max.   :1.333   Max.   :57

mining info:
         data ntransactions support confidence
 trans_retail          4000    0.01       0.01
```

```
# The most populars products
retpro_sup <- inspect(head(sort(rules_retpro, by = "support"), 10))
```

```
    lhs                            rhs                          support
[1] {CYBERPOWER Gamer Desktop} => {Apple Earpods}             0.01425
[2] {Apple Earpods}            => {CYBERPOWER Gamer Desktop} 0.01425
    confidence lift     count
[1] 0.1676471  1.333177 57
[2] 0.1133201  1.333177 57
```

```
# Items that have high chances of being bought together
retpro_con <- inspect(head(sort(rules_retpro, by = "confidence"), 10))
```

```
    lhs                            rhs                          support
[1] {CYBERPOWER Gamer Desktop} => {Apple Earpods}             0.01425
[2] {Apple Earpods}            => {CYBERPOWER Gamer Desktop} 0.01425
    confidence lift     count
[1] 0.1676471  1.333177 57
[2] 0.1133201  1.333177 57
```

```
# Lift
retpro_lift <- inspect(head(sort(rules_retpro, by = "lift"), 10))
```

```
    lhs                            rhs                          support
[1] {Apple Earpods}            => {CYBERPOWER Gamer Desktop} 0.01425
[2] {CYBERPOWER Gamer Desktop} => {Apple Earpods}             0.01425
    confidence lift     count
[1] 0.1133201  1.333177 57
[2] 0.1676471  1.333177 57
```

Rules for categories in retailers transactions

```
trans_retcat <- aggregate(trans_retail, by = "category")
rules_retcat <- apriori(trans_retcat, parameter = list(supp = 0.01, conf = 0.01,
    minlen = 2, maxlen = 4))
rules_retcat <- rules_retcat[!is.redundant(rules_retcat)]
```

```
summary(rules_retcat)
```

```
set of 40 rules

rule length distribution (lhs + rhs):sizes
 2
40

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
      2       2       2       2       2       2

summary of quality measures:
```

```
    support           confidence          lift            count
 Min.   :0.01025   Min.   :0.03275   Min.   :0.5645   Min.   : 41
 1st Qu.:0.01250   1st Qu.:0.08498   1st Qu.:0.6138   1st Qu.: 50
 Median :0.01675   Median :0.12546   Median :0.7535   Median : 67
 Mean   :0.02275   Mean   :0.15262   Mean   :0.8038   Mean   : 91
 3rd Qu.:0.03625   3rd Qu.:0.21769   3rd Qu.:0.9706   3rd Qu.:145
 Max.   :0.05375   Max.   :0.35602   Max.   :1.2459   Max.   :215

mining info:
         data ntransactions support confidence
 trans_retcat          4000    0.01       0.01
```

```r
# The most populars products
retcat_sup <- inspect(head(sort(rules_retcat, by = "support"), 10))
```

```
     lhs                    rhs                 support confidence
[1]  {Monitors}          => {Desktop}           0.05375 0.3272451
[2]  {Desktop}           => {Monitors}          0.05375 0.1881015
[3]  {Computer Mice}     => {Desktop}           0.03825 0.2875940
[4]  {Desktop}           => {Computer Mice}     0.03825 0.1338583
[5]  {Monitors}          => {Laptops}           0.03725 0.2267884
[6]  {Laptops}           => {Monitors}          0.03725 0.1190096
[7]  {Keyboard}          => {Desktop}           0.03625 0.2859961
[8]  {Desktop}           => {Keyboard}          0.03625 0.1268591
[9]  {Active Headphones} => {Laptops}           0.03625 0.2449324
[10] {Laptops}           => {Active Headphones} 0.03625 0.1158147
     lift      count
[1]  1.1452145 215
[2]  1.1452145 215
[3]  1.0064531 153
[4]  1.0064531 153
[5]  0.7245637 149
[6]  0.7245637 149
[7]  1.0008611 145
[8]  1.0008611 145
[9]  0.7825317 145
[10] 0.7825317 145
```

```r
# Items that have high chances of being bought together
retcat_con <- inspect(head(sort(rules_retcat, by = "confidence"), 10))
```

```
     lhs                           rhs       support confidence lift
[1]  {Accessories}              => {Desktop} 0.01700 0.3560209  1.2459176
[2]  {Monitors}                 => {Desktop} 0.05375 0.3272451  1.1452145
[3]  {Mouse and Keyboard Combo} => {Laptops} 0.02275 0.3204225  1.0237142
[4]  {Computer Cords}           => {Laptops} 0.01150 0.3006536  0.9605546
[5]  {Computer Mice}            => {Desktop} 0.03825 0.2875940  1.0064531
[6]  {Keyboard}                 => {Desktop} 0.03625 0.2859961  1.0008611
[7]  {Computer Headphones}      => {Desktop} 0.01250 0.2577320  0.9019491
[8]  {Active Headphones}        => {Laptops} 0.03625 0.2449324  0.7825317
[9]  {Active Headphones}        => {Desktop} 0.03625 0.2449324  0.8571564
[10] {Monitors}                 => {Laptops} 0.03725 0.2267884  0.7245637
     count
```

```
[1]   68
[2]  215
[3]   91
[4]   46
[5]  153
[6]  145
[7]   50
[8]  145
[9]  145
[10] 149
```

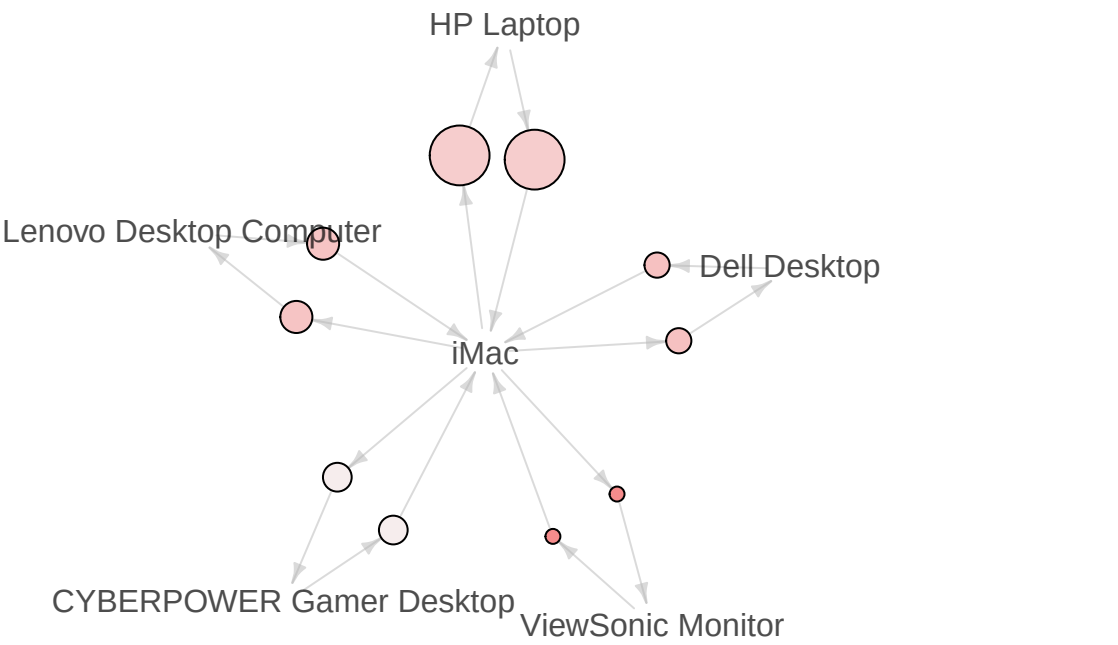
```
# Lift
retcat_lift <- inspect(head(sort(rules_retcat, by = "lift"), 10))
```

```
     lhs                           rhs                           support
[1]  {Accessories}              => {Desktop}                     0.01700
[2]  {Desktop}                  => {Accessories}                 0.01700
[3]  {Desktop}                  => {Monitors}                    0.05375
[4]  {Monitors}                 => {Desktop}                     0.05375
[5]  {Mouse and Keyboard Combo} => {Laptops}                     0.02275
[6]  {Laptops}                  => {Mouse and Keyboard Combo}    0.02275
[7]  {Computer Mice}            => {Desktop}                     0.03825
[8]  {Desktop}                  => {Computer Mice}               0.03825
[9]  {Keyboard}                 => {Desktop}                     0.03625
[10] {Desktop}                  => {Keyboard}                    0.03625
     confidence lift     count
[1]  0.35602094 1.245918  68
[2]  0.05949256 1.245918  68
[3]  0.18810149 1.145215 215
[4]  0.32724505 1.145215 215
[5]  0.32042254 1.023714  91
[6]  0.07268371 1.023714  91
[7]  0.28759398 1.006453 153
[8]  0.13385827 1.006453 153
[9]  0.28599606 1.000861 145
[10] 0.12685914 1.000861 145
```

Rules visualizations

```
methods <- c("graph", "scatterplot")
for (i in methods) {
    plot(rules_corpro, method = i, control = list(type = "items"), max = 10,
        main = paste(capitalize(i), "of rules of products bought by corporates"))
    plot(rules_corcat, method = i, control = list(type = "items"), max = 10,
        main = paste(capitalize(i), "of rules of categories bought by corporates"))
    plot(rules_retpro, method = i, control = list(type = "items"), max = 10,
        main = paste(capitalize(i), "of rules of products bought by retailers"))
    plot(rules_retcat, method = i, control = list(type = "items"), max = 10,
        main = paste(capitalize(i), "of rules of categories bought by retailers"))

}
```

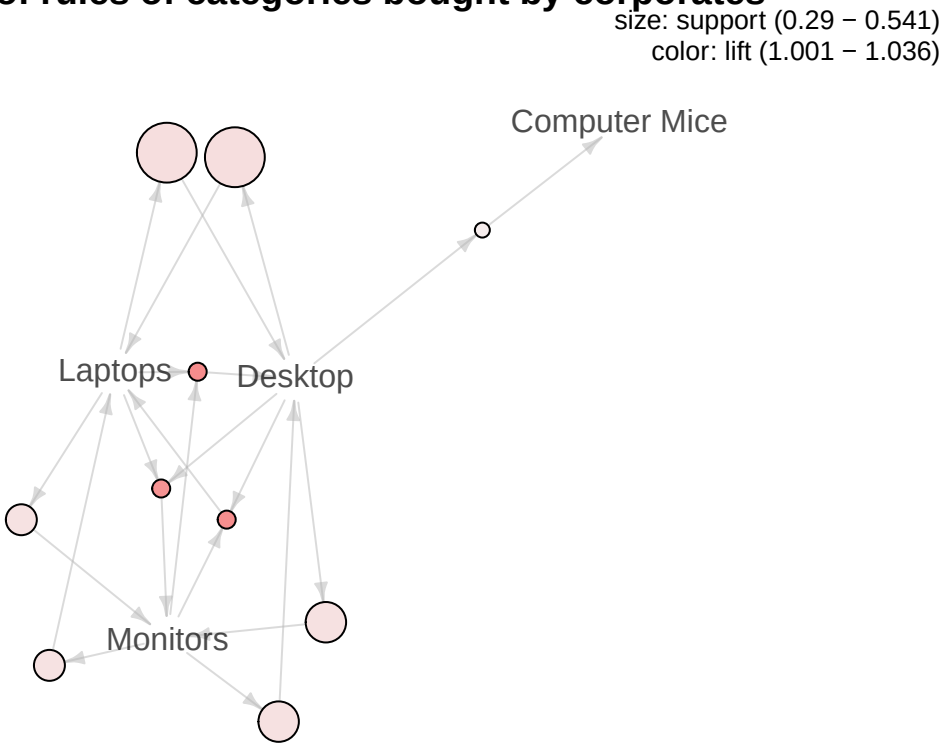
```
Available control parameters (with default values):
main     =  Graph for 10 rules
nodeColors   =  c("#66CC6680", "#9999CC80")
nodeCol  =  c("#EE0000FF", "#EE0303FF", "#EE0606FF", "#EE0909FF", "#EE0C0CFF", "#EE0F0FFF", "#EE1212FF"
edgeCol  =  c("#474747FF", "#494949FF", "#4B4B4BFF", "#4D4D4DFF", "#4F4F4FFF", "#515151FF", "#535353FF"
alpha    =  0.5
cex  =  1
itemLabels   =  TRUE
labelCol     =  #000000B3
measureLabels    =  FALSE
precision    =  3
layout   =  NULL
layoutParams     =  list()
arrowSize    =  0.5
engine   =  igraph
plot     =  TRUE
plot_options     =  list()
max  =  100
verbose  =  FALSE
```




```
Available control parameters (with default values):
main     =  Graph for 10 rules
nodeColors   =  c("#66CC6680", "#9999CC80")
nodeCol  =  c("#EE0000FF", "#EE0303FF", "#EE0606FF", "#EE0909FF", "#EE0C0CFF", "#EE0F0FFF", "#EE1212FF"
edgeCol  =  c("#474747FF", "#494949FF", "#4B4B4BFF", "#4D4D4DFF", "#4F4F4FFF", "#515151FF", "#535353FF"
alpha    =  0.5
cex  =  1
itemLabels   =  TRUE
labelCol     =  #000000B3
```

```
measureLabels    =  FALSE
precision    =  3
layout   =  NULL
layoutParams     =  list()
arrowSize    =  0.5
engine   =  igraph
plot     =  TRUE
plot_options     =  list()
max  =  100
verbose  =  FALSE
```

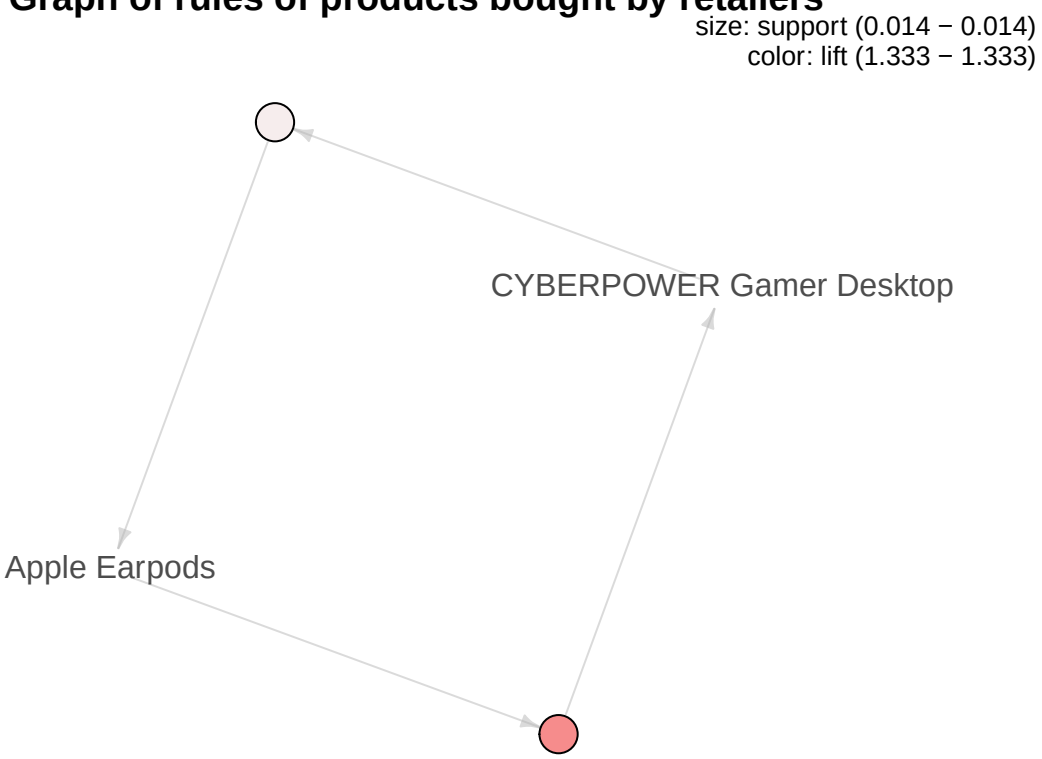
Graph of rules of categories bought by corporates




```
Available control parameters (with default values):
main     =  Graph for 2 rules
nodeColors   =  c("#66CC6680", "#9999CC80")
nodeCol  =  c("#EE0000FF", "#EE0303FF", "#EE0606FF", "#EE0909FF", "#EE0C0CFF", "#EE0F0FFF", "#EE1212FF"
edgeCol  =  c("#474747FF", "#494949FF", "#4B4B4BFF", "#4D4D4DFF", "#4F4F4FFF", "#515151FF", "#535353FF"
alpha    =  0.5
cex  =  1
itemLabels   =  TRUE
labelCol     =  #000000B3
measureLabels    =  FALSE
precision    =  3
layout   =  NULL
layoutParams     =  list()
arrowSize    =  0.5
engine   =  igraph
plot     =  TRUE
plot_options     =  list()
max  =  100
```

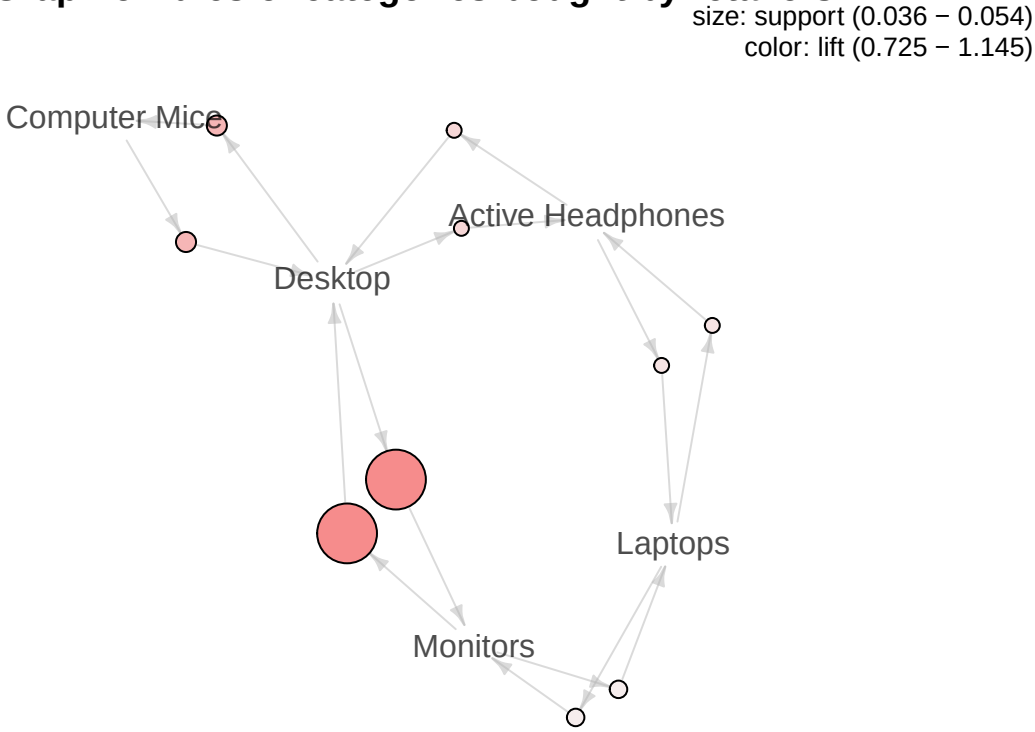
```
verbose  =  FALSE
```

Graph of rules of products bought by retailers



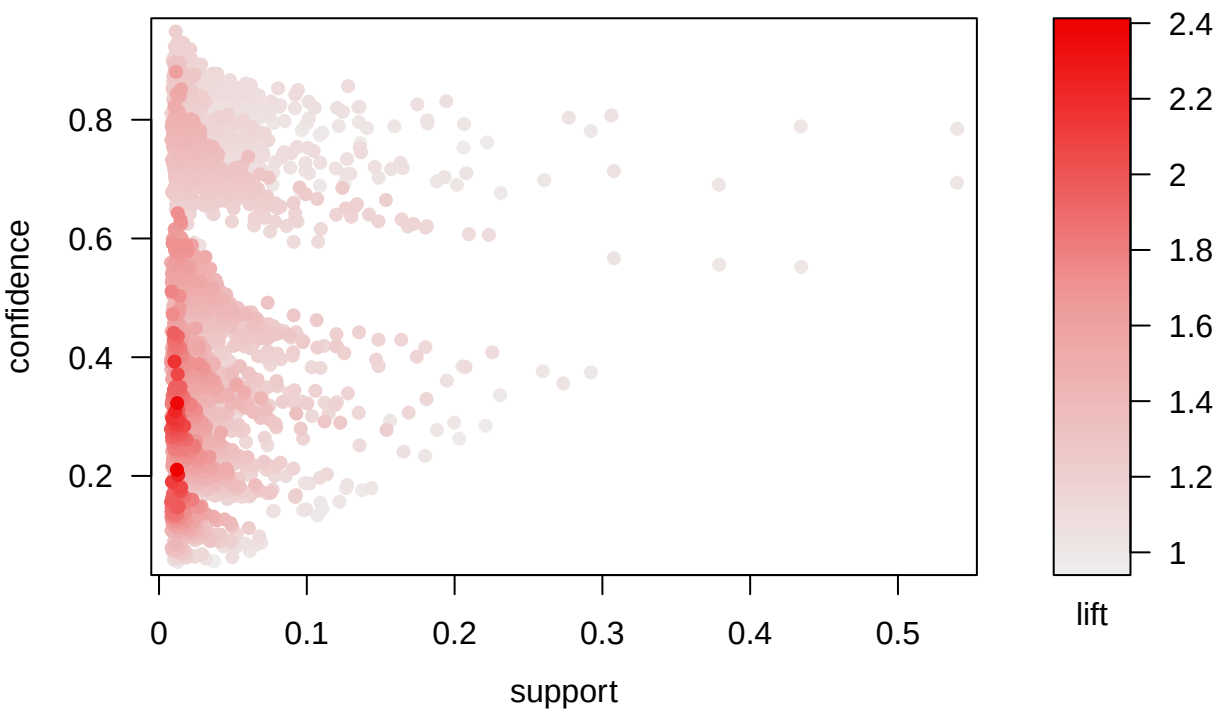

```
Available control parameters (with default values):
main     =  Graph for 10 rules
nodeColors   =  c("#66CC6680", "#9999CC80")
nodeCol  =  c("#EE0000FF", "#EE0303FF", "#EE0606FF", "#EE0909FF", "#EE0C0CFF", "#EE0F0FFF", "#EE1212FF"
edgeCol  =  c("#474747FF", "#494949FF", "#4B4B4BFF", "#4D4D4DFF", "#4F4F4FFF", "#515151FF", "#535353FF"
alpha    =  0.5
cex  =  1
itemLabels   =  TRUE
labelCol     =  #000000B3
measureLabels    =  FALSE
precision    =  3
layout   =  NULL
layoutParams     =  list()
arrowSize    =  0.5
engine   =  igraph
plot     =  TRUE
plot_options     =  list()
max  =  100
verbose  =  FALSE
```

Graph of rules of categories bought by retailers

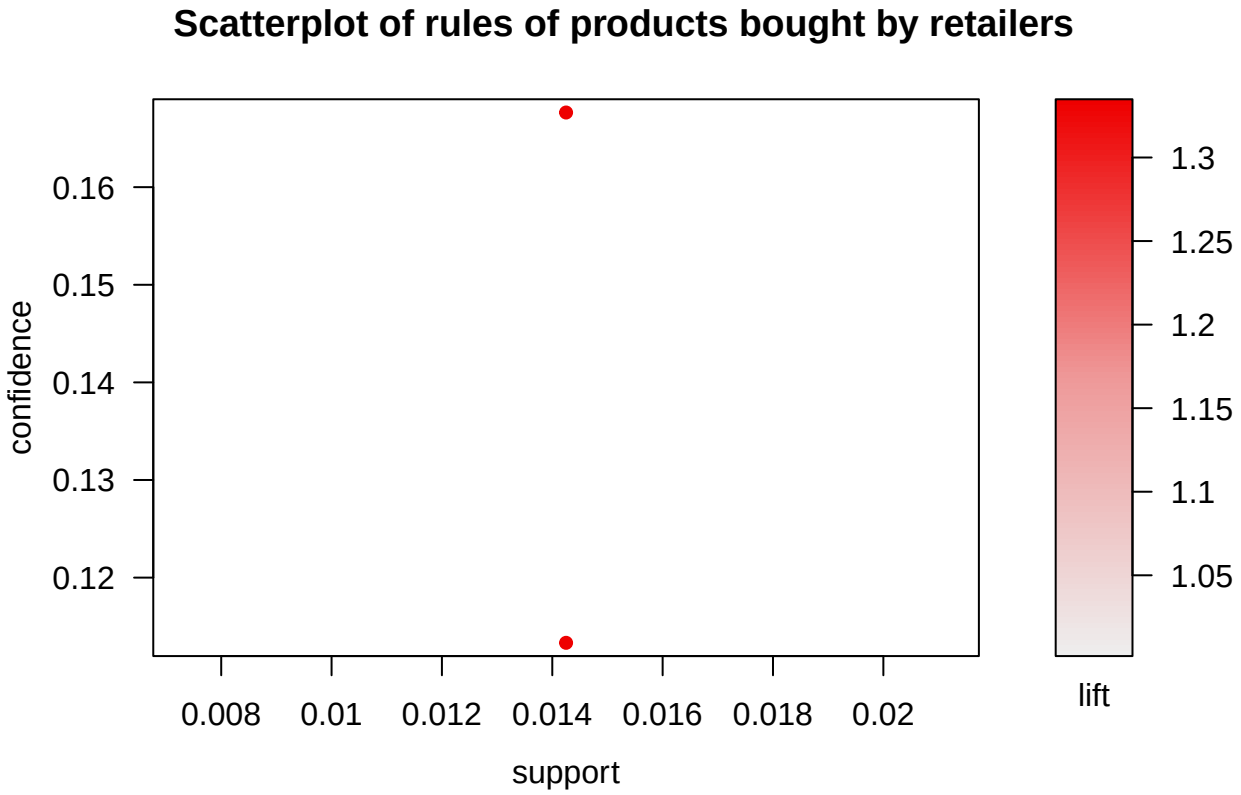



```
Available control parameters (with default values):
main     =  Scatter plot for 1607 rules
engine   =  default
pch  =  19
cex  =  0.5
xlim     =  NULL
ylim     =  NULL
zlim     =  NULL
alpha    =  NULL
col  =  c("#EE0000FF", "#EE0303FF", "#EE0606FF", "#EE0909FF", "#EE0C0CFF", "#EE0F0FFF", "#EE1212FF", "#E
newpage  =  TRUE
jitter   =  NA
verbose  =  FALSE
```

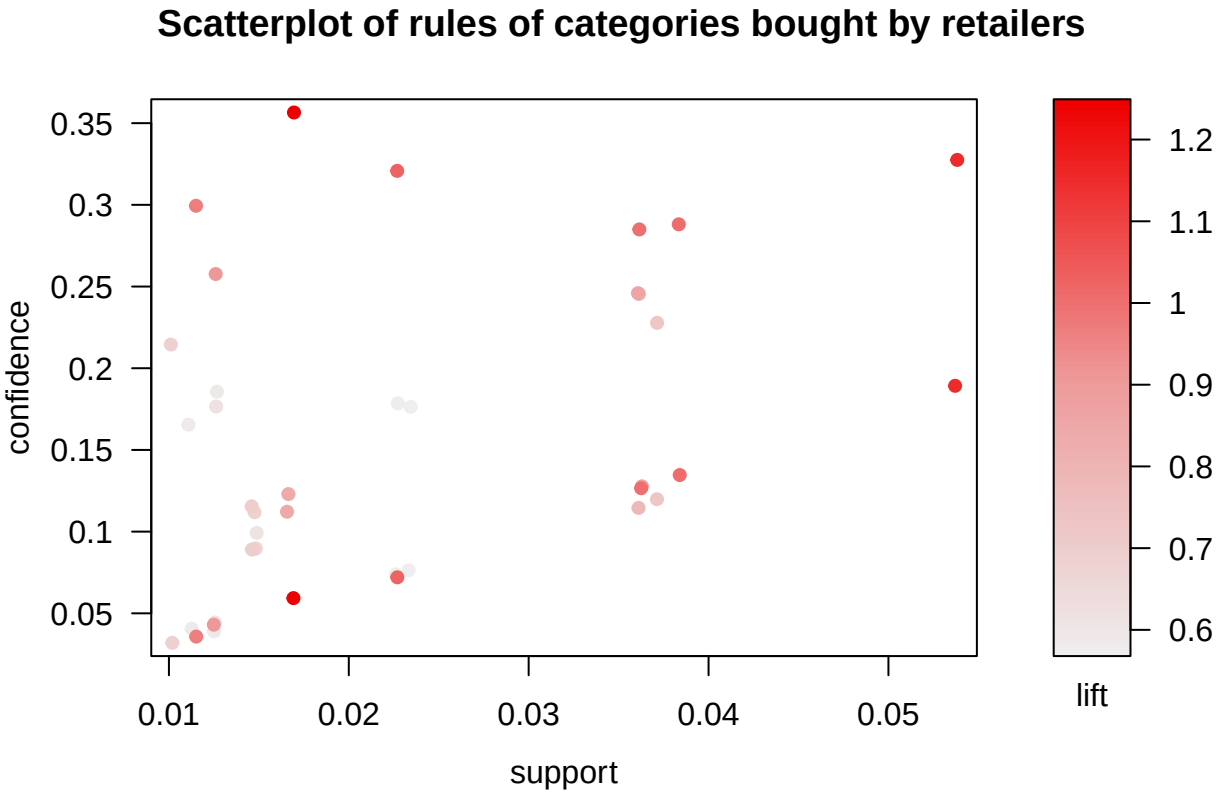

```
Available control parameters (with default values):
main     =  Scatter plot for 3599 rules
engine   =  default
pch  =  19
cex  =  0.5
xlim     =  NULL
ylim     =  NULL
zlim     =  NULL
alpha    =  NULL
col  =  c("#EE0000FF", "#EE0303FF", "#EE0606FF", "#EE0909FF", "#EE0C0CFF", "#EE0F0FFF", "#EE1212FF", "#E
newpage  =  TRUE
jitter   =  NA
verbose  =  FALSE
```




```
Available control parameters (with default values):
main     =  Scatter plot for 2 rules
engine   =  default
pch  =  19
cex  =  0.5
xlim     =  NULL
ylim     =  NULL
zlim     =  NULL
alpha    =  NULL
col  =  c("#EE0000FF", "#EE0303FF", "#EE0606FF", "#EE0909FF", "#EE0C0CFF", "#EE0F0FFF", "#EE1212FF", "#
newpage  =  TRUE
jitter   =  NA
verbose  =  FALSE
```




```
Available control parameters (with default values):
main     =  Scatter plot for 40 rules
engine   =  default
pch  =  19
cex  =  0.5
xlim     =  NULL
ylim     =  NULL
zlim     =  NULL
alpha    =  NULL
col  =  c("#EE0000FF", "#EE0303FF", "#EE0606FF", "#EE0909FF", "#EE0C0CFF", "#EE0F0FFF", "#EE1212FF", "#
newpage  =  TRUE
jitter   =  NA
verbose  =  FALSE
```


Scatterplot of rules of categories bought by retailers
confidence
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0.01
0.02
0.03
0.04
0.05
support
lift
1.2
1.1
1
0.9
0.8
0.7
0.6