ApplicationOfRInMarketing.r

MATEO

2023-02-21

```
############################### DATA PREPARATION ###############################
# install.packages(c("ggplot2", "dplyr", "tidyr", "RColorBrewer"))
library("ggplot2")
```

```
## Warning: package 'ggplot2' was built under R version 4.1.3
```

```
library("dplyr")
```

```
## Warning: package 'dplyr' was built under R version 4.1.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library("tidyr")
```

```
## Warning: package 'tidyr' was built under R version 4.1.3
```

```
library("RColorBrewer")
```

```
## Warning: package 'RColorBrewer' was built under R version 4.1.3
```

```
# Load data from CSV file and check its structure
rawSalesData <- read.csv("SuperstoreSalesTraining.csv", na.strings = "", stringsAsFactors = TRUE)
str(rawSalesData)
```

```
## 'data.frame':    16798 obs. of  26 variables:
##  $ Row                : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Order.Priority     : Factor w/ 5 levels "Critical","High",..: 2 5 1 3 3 3 1 1 1 3 ...
##  $ Order.Date         : Factor w/ 1427 levels "01/01/2010","01/01/2011",..: 1 1 48 48 48 48 48 48 48 48 ...
##  $ Order              : int  28774 88028 9285 37537 37537 37537 44069 44069 89083 87946 ...
##  $ Discount           : Factor w/ 17 levels "0%","1%","10%",..: 3 15 13 1 14 12 16 15 13 12 ...
##  $ Unit.Price         : num  6 96 41 292 101 155 9 15 41 155 ...
##  $ Order.Quantity     : int  32 2 3 4 43 32 16 43 1 8 ...
##  $ Sales              : num  173 177 116 1168 4039 ...
##  $ Profit             : num  106.4 45.6 33.9 605.1 2647.7 ...
##  $ Shipping.Cost      : num  5 35 3 49 45 7 2 2 3 7 ...
##  $ Product.Base.Margin: Factor w/ 61 levels "14.50%","3.60%",..: 43 25 7 31 44 34 13 10 7 34 ...
##  $ Department         : Factor w/ 3 levels "Furniture","Office Supplies",..: 2 2 2 1 1 2 2 2 2 2 ...
##  $ Container          : Factor w/ 7 levels "Jumbo Box","Jumbo Drum",..: 5 3 5 2 2 5 7 7 5 5 ...
##  $ Category           : Factor w/ 17 levels "Appliances","Binders and Binder Accessories",..: 15 15 2 4 4 15 11 11 2 15
...
##  $ Item               : Factor w/ 1263 levels "\"While you Were Out\" Message Book, One Form per Page",..: 862 921 229 60
6 625 395 276 1115 229 395 ...
##  $ Customer.Segment   : Factor w/ 4 levels "Consumer","Corporate",..: 4 3 1 2 2 2 1 1 1 2 ...
##  $ Customer_ID        : int  1656 2211 949 68 68 68 1154 1154 950 67 ...
##  $ Customer.Name      : Factor w/ 3403 levels "Aaron Davies Bruce",..: 1686 133 981 2946 2946 2946 2219 2219 1430 916 ...
##  $ Region             : Factor w/ 4 levels "AsiaPac","EMEA",..: 1 4 4 4 4 4 1 1 4 4 ...
##  $ State              : Factor w/ 149 levels "?saka","Addis Ababa",..: 22 66 19 85 85 85 1 1 71 19 ...
##  $ Country...Region   : Factor w/ 50 levels "Algeria","Argentina",..: 14 49 49 49 49 49 25 25 49 49 ...
##  $ City               : Factor w/ 1523 levels "Aberdeen","Abidjan",..: 1327 136 760 916 916 916 992 992 1100 893 ...
##  $ Postal.Code        : int  NA 20715 90049 10177 10177 10177 NA NA 55372 94559 ...
##  $ Ship.Date          : Factor w/ 1459 levels "01/01/2011","01/01/2012",..: 48 97 144 48 144 385 144 48 144 385 ...
##  $ Ship.Mode          : Factor w/ 3 levels "Delivery Truck",..: 3 2 3 1 1 3 2 3 3 3 ...
##  $ SubRegion          : Factor w/ 5 levels "Canada
"| __truncated__,..: NA 3 5 3 3 3 NA NA 2 5 ...
```

```
summary(rawSalesData)
```

```
##       Row              Order.Priority      Order.Date        Order      
##  Min.   :    1   Critical     :3216   28/03/2013:   47   Min.   :    3  
##  1st Qu.: 4200   High         :3536   15/09/2012:   40   1st Qu.:29858  
##  Median : 8400   Low          :3440   05/01/2013:   37   Median :72896  
##  Mean   : 8400   Medium       :3262   18/10/2013:   36   Mean   :59335  
##  3rd Qu.:12599   Not Specified:3344   19/11/2012:   34   3rd Qu.:88699  
##  Max.   :16798                        21/07/2013:   34   Max.   :91591  
##                                       (Other)   :16570                  
##     Discount      Unit.Price      Order.Quantity       Sales         
##  1%     :1599   Min.   :   1.00   Min.   :  1.00   Min.   :    0.90  
##  5%     :1564   1st Qu.:   6.00   1st Qu.:  8.00   1st Qu.:   90.11  
##  3%     :1547   Median :  21.00   Median : 16.00   Median :  336.00  
##  9%     :1543   Mean   :  89.33   Mean   : 26.22   Mean   : 1790.07  
##  4%     :1525   3rd Qu.:  86.00   3rd Qu.: 38.00   3rd Qu.: 1391.16  
##  2%     :1518   Max.   :6783.00   Max.   :288.00   Max.   :99130.12  
##  (Other):7502                                                        
##      Profit         Shipping.Cost    Product.Base.Margin           Department  
##  Min.   :-4301.08   Min.   :  0.00   37.00% : 1474       Furniture      :3448  
##  1st Qu.:   28.52   1st Qu.:  3.00   38.00% : 1266       Office Supplies:9220  
##  Median :  133.65   Median :  6.00   36.00% : 1190       Technology     :4130  
##  Mean   :  882.15   Mean   : 12.86   59.00% :  962                             
##  3rd Qu.:  655.66   3rd Qu.: 14.00   56.00% :  918                             
##  Max.   :60250.64   Max.   :165.00   57.00% :  918                             
##                                      (Other):10070                             
##       Container                             Category   
##  Jumbo Box :1064   Paper                        :2450  
##  Jumbo Drum:1248   Binders and Binder Accessories:1830  
##  Large Box : 812   Telephones and Communication  :1766  
##  Medium Box: 732   Office Furnishings            :1576  
##  Small Box :8694   Computer Peripherals          :1516  
##  Small Pack:1912   Pens & Art Supplies           :1266  
##  Wrap Bag  :2336   (Other)                       :6394  
##                                                                    Item      
##  Global High-Back Leather Tilter, Burgundy                        :   48  
##  Bevis 36 x 72 Conference Tables                                  :   44  
##  BoxOffice By Design Rectangular and Half-Moon Meeting Room Tables:   44  
##  Fiskars® Softgrip Scissors                                       :   44  
##  Master Giant Foot® Doorstop, Safety Yellow                       :   44  
##  Wilson Jones Hanging View Binder, White, 1"                      :   42  
##  (Other)                                                          :16532  
##        Customer.Segment  Customer_ID            Customer.Name  
##  Consumer      :3298    Min.   :   1   Rosemary Hedrick:   41  
##  Corporate     :6152    1st Qu.: 912   Sylvia Barr     :   38  
##  Home Office   :4064    Median :1778   Jason Fink      :   35  
##  Small Business:3284    Mean   :1754   Courtney McBride:   33  
##                         3rd Qu.:2593   Annie Rouse     :   30  
##                         Max.   :3403   Kevin Erickson  :   29  
##                                        (Other)         :16592  
##            Region                   State                       Country...Region
##  AsiaPac      :3802   California     : 1021   United States of America:9426  
##  EMEA         :1894   Texas          :  646   China                   :1257  
##  Latam        :1620   Illinois       :  584   India                   : 746  
##  North America:9482   New York       :  574   Brazil                  : 672  
##                       Florida        :  522   Japan                   : 507  
##                       Guangdong Sheng:  417   Mexico                  : 388  
##                       (Other)        :13034   (Other)                 :3802  
##            City         Postal.Code         Ship.Date              Ship.Mode    
##  Guangzhou   :  357   Min.   : 1001   21/05/2012:   38   Delivery Truck: 2292  
##  Buenos Aires:  341   1st Qu.:28352   09/05/2013:   35   Express Air   : 1966  
##  Seoul       :  292   Median :53081   27/05/2013:   34   Regular Air   :12540  
##  Tokyo       :  286   Mean   :52312   04/10/2013:   33                         
##  Paris       :  248   3rd Qu.:77530   30/03/2013:   33                         
##  Beijing     :  245   Max.   :99362   02/06/2013:   31                         
##  (Other)     :15029   NA's   :6985    (Other)   :16594                         
##                                                                                
SubRegion
##  Canada
:  56
##  Central
:2899
##  East
:2289
##  South
:1954
##  West
:2284
##  NA's
:7316
##
```

```
# Remove unnecessary variables (rawSalesData$Row)
salesData <- subset(rawSalesData, select=-c(Row))
str(salesData)
```

```
## 'data.frame':    16798 obs. of  25 variables:
##  $ Order.Priority     : Factor w/ 5 levels "Critical","High",..: 2 5 1 3 3 3 1 1 1 3 ...
##  $ Order.Date         : Factor w/ 1427 levels "01/01/2010","01/01/2011",..: 1 1 48 48 48 48 48 48 48 48 ...
##  $ Order              : int  28774 88028 9285 37537 37537 37537 44069 44069 89083 87946 ...
##  $ Discount           : Factor w/ 17 levels "0%","1%","10%",..: 3 15 13 1 14 12 16 15 13 12 ...
##  $ Unit.Price         : num  6 96 41 292 101 155 9 15 41 155 ...
##  $ Order.Quantity     : int  32 2 3 4 43 32 16 43 1 8 ...
##  $ Sales              : num  173 177 116 1168 4039 ...
##  $ Profit             : num  106.4 45.6 33.9 605.1 2647.7 ...
##  $ Shipping.Cost      : num  5 35 3 49 45 7 2 2 3 7 ...
##  $ Product.Base.Margin: Factor w/ 61 levels "14.50%","3.60%",..: 43 25 7 31 44 34 13 10 7 34 ...
##  $ Department         : Factor w/ 3 levels "Furniture","Office Supplies",..: 2 2 2 1 1 2 2 2 2 2 ...
##  $ Container          : Factor w/ 7 levels "Jumbo Box","Jumbo Drum",..: 5 3 5 2 2 5 7 7 5 5 ...
##  $ Category           : Factor w/ 17 levels "Appliances","Binders and Binder Accessories",..: 15 15 2 4 4 15 11 11 2 15
...
##  $ Item               : Factor w/ 1263 levels "\"While you Were Out\" Message Book, One Form per Page",..: 862 921 229 60
6 625 395 276 1115 229 395 ...
##  $ Customer.Segment   : Factor w/ 4 levels "Consumer","Corporate",..: 4 3 1 2 2 2 1 1 1 2 ...
##  $ Customer_ID        : int  1656 2211 949 68 68 68 1154 1154 950 67 ...
##  $ Customer.Name      : Factor w/ 3403 levels "Aaron Davies Bruce",..: 1686 133 981 2946 2946 2946 2219 2219 1430 916 ...
##  $ Region             : Factor w/ 4 levels "AsiaPac","EMEA",..: 1 4 4 4 4 4 1 1 4 4 ...
##  $ State              : Factor w/ 149 levels "?saka","Addis Ababa",..: 22 66 19 85 85 85 1 1 71 19 ...
##  $ Country...Region   : Factor w/ 50 levels "Algeria","Argentina",..: 14 49 49 49 49 49 25 25 49 49 ...
##  $ City               : Factor w/ 1523 levels "Aberdeen","Abidjan",..: 1327 136 760 916 916 916 992 992 1100 893 ...
##  $ Postal.Code        : int  NA 20715 90049 10177 10177 10177 NA NA 55372 94559 ...
##  $ Ship.Date          : Factor w/ 1459 levels "01/01/2011","01/01/2012",..: 48 97 144 48 144 385 144 48 144 385 ...
##  $ Ship.Mode          : Factor w/ 3 levels "Delivery Truck",..: 3 2 3 1 1 3 2 3 3 3 ...
##  $ SubRegion          : Factor w/ 5 levels "Canada
"| __truncated__,..: NA 3 5 3 3 3 NA NA 2 5 ...
```

```
summary(salesData)
```

```
##        Order.Priority      Order.Date        Order          Discount    
##  Critical     :3216   28/03/2013:   47   Min.   :    3   1%     :1599  
##  High         :3536   15/09/2012:   40   1st Qu.:29858   5%     :1564  
##  Low          :3440   05/01/2013:   37   Median :72896   3%     :1547  
##  Medium       :3262   18/10/2013:   36   Mean   :59335   9%     :1543  
##  Not Specified:3344   19/11/2012:   34   3rd Qu.:88699   4%     :1525  
##                       21/07/2013:   34   Max.   :91591   2%     :1518  
##                       (Other)   :16570                   (Other):7502  
##    Unit.Price      Order.Quantity       Sales             Profit        
##  Min.   :   1.00   Min.   :  1.00   Min.   :    0.90   Min.   :-4301.08  
##  1st Qu.:   6.00   1st Qu.:  8.00   1st Qu.:   90.11   1st Qu.:   28.52  
##  Median :  21.00   Median : 16.00   Median :  336.00   Median :  133.65  
##  Mean   :  89.33   Mean   : 26.22   Mean   : 1790.07   Mean   :  882.15  
##  3rd Qu.:  86.00   3rd Qu.: 38.00   3rd Qu.: 1391.16   3rd Qu.:  655.66  
##  Max.   :6783.00   Max.   :288.00   Max.   :99130.12   Max.   :60250.64  
##                                                                          
##  Shipping.Cost    Product.Base.Margin           Department        Container    
##  Min.   :  0.00   37.00% : 1474       Furniture      :3448   Jumbo Box :1064  
##  1st Qu.:  3.00   38.00% : 1266       Office Supplies:9220   Jumbo Drum:1248  
##  Median :  6.00   36.00% : 1190       Technology     :4130   Large Box : 812  
##  Mean   : 12.86   59.00% :  962                              Medium Box: 732  
##  3rd Qu.: 14.00   56.00% :  918                              Small Box :8694  
##  Max.   :165.00   57.00% :  918                              Small Pack:1912  
##                   (Other):10070                              Wrap Bag  :2336  
##                           Category    
##  Paper                        :2450  
##  Binders and Binder Accessories:1830  
##  Telephones and Communication  :1766  
##  Office Furnishings            :1576  
##  Computer Peripherals          :1516  
##  Pens & Art Supplies           :1266  
##  (Other)                       :6394  
##                                                                        Item      
##  Global High-Back Leather Tilter, Burgundy                        :   48  
##  Bevis 36 x 72 Conference Tables                                  :   44  
##  BoxOffice By Design Rectangular and Half-Moon Meeting Room Tables:   44  
##  Fiskars® Softgrip Scissors                                       :   44  
##  Master Giant Foot® Doorstop, Safety Yellow                       :   44  
##  Wilson Jones Hanging View Binder, White, 1"                      :   42  
##  (Other)                                                          :16532  
##        Customer.Segment  Customer_ID            Customer.Name  
##  Consumer      :3298    Min.   :   1   Rosemary Hedrick:   41  
##  Corporate     :6152    1st Qu.: 912   Sylvia Barr     :   38  
##  Home Office   :4064    Median :1778   Jason Fink      :   35  
##  Small Business:3284    Mean   :1754   Courtney McBride:   33  
##                         3rd Qu.:2593   Annie Rouse     :   30  
##                         Max.   :3403   Kevin Erickson  :   29  
##                                        (Other)         :16592  
##            Region                 State                       Country...Region
##  AsiaPac      :3802   California     : 1021   United States of America:9426  
##  EMEA         :1894   Texas          :  646   China                   :1257  
##  Latam        :1620   Illinois       :  584   India                   : 746  
##  North America:9482   New York       :  574   Brazil                  : 672  
##                       Florida        :  522   Japan                   : 507  
##                       Guangdong Sheng:  417   Mexico                  : 388  
##                       (Other)        :13034   (Other)                 :3802  
##            City        Postal.Code         Ship.Date              Ship.Mode    
##  Guangzhou   :  357   Min.   : 1001   21/05/2012:   38   Delivery Truck: 2292  
##  Buenos Aires:  341   1st Qu.:28352   09/05/2013:   35   Express Air   : 1966  
##  Seoul       :  292   Median :53081   27/05/2013:   34   Regular Air   :12540  
##  Tokyo       :  286   Mean   :52312   04/10/2013:   33                         
##  Paris       :  248   3rd Qu.:77530   30/03/2013:   33                         
##  Beijing     :  245   Max.   :99362   02/06/2013:   31                         
##  (Other)     :15029   NA's   :6985    (Other)   :16594                         
##                                                                                
SubRegion
##  Canada
:  56
##  Central
:2899
##  East
:2289
##  South
:1954
##  West
:2284
##  NA's
:7316
##
```

```
# Filter NA values
colnames(salesData) # All columns
```

```
##  [1] "Order.Priority"      "Order.Date"          "Order"
##  [4] "Discount"            "Unit.Price"          "Order.Quantity"
##  [7] "Sales"               "Profit"              "Shipping.Cost"
## [10] "Product.Base.Margin" "Department"          "Container"
## [13] "Category"            "Item"                "Customer.Segment"
## [16] "Customer_ID"         "Customer.Name"       "Region"
## [19] "State"               "Country...Region"    "City"
## [22] "Postal.Code"         "Ship.Date"           "Ship.Mode"
## [25] "SubRegion"
```

```
colnames(salesData[, colSums(is.na(salesData)) == 0]) # Non NA columns
```

```
##  [1] "Order.Priority"      "Order.Date"          "Order"
##  [4] "Discount"            "Unit.Price"          "Order.Quantity"
##  [7] "Sales"               "Profit"              "Shipping.Cost"
## [10] "Product.Base.Margin" "Department"          "Container"
## [13] "Category"            "Item"                "Customer.Segment"
## [16] "Customer_ID"         "Customer.Name"       "Region"
## [19] "State"               "Country...Region"    "City"
## [22] "Ship.Date"           "Ship.Mode"
```

```
colnames(salesData[, colSums(is.na(salesData)) > 0])  # NA columns ("Postal.Code", "SubRegion")
```

```
## [1] "Postal.Code" "SubRegion"
```

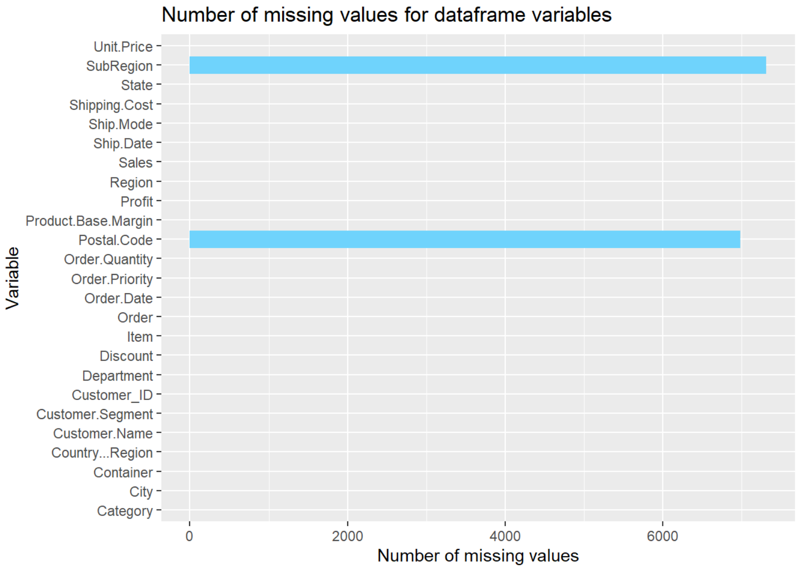
```
sum(is.na(salesData$Postal.Code)) # 6985 NA values
```

```
## [1] 6985
```

```
sum(is.na(salesData$SubRegion))   # 7316 NA values
```

```
## [1] 7316
```

```
lapply(salesData, function(l) sum(is.na(l))) %>%
  data.frame() %>%
  pivot_longer(names_to = "columns", cols = names(.), values_to = "value") %>%
  ggplot(aes(x = columns, y = value)) +
  geom_bar(stat = "identity", fill = "#6FD3FC") +
  coord_flip() +
  labs(x = "Variable", y = "Number of missing values",
       title = "Number of missing values for dataframe variables")
```



```
salesData <- salesData[, colSums(is.na(salesData)) == 0]
str(salesData)
```

```
## 'data.frame':    16798 obs. of  23 variables:
##  $ Order.Priority     : Factor w/ 5 levels "Critical","High",..: 2 5 1 3 3 3 1 1 1 3 ...
##  $ Order.Date         : Factor w/ 1427 levels "01/01/2010","01/01/2011",..: 1 1 48 48 48 48 48 48 48 48 ...
##  $ Order              : int  28774 88028 9285 37537 37537 37537 44069 44069 89083 87946 ...
##  $ Discount           : Factor w/ 17 levels "0%","1%","10%",..: 3 15 13 1 14 12 16 15 13 12 ...
##  $ Unit.Price         : num  6 96 41 292 101 155 9 15 41 155 ...
##  $ Order.Quantity     : int  32 2 3 4 43 32 16 43 1 8 ...
##  $ Sales              : num  173 177 116 1168 4039 ...
##  $ Profit             : num  106.4 45.6 33.9 605.1 2647.7 ...
##  $ Shipping.Cost      : num  5 35 3 49 45 7 2 2 3 7 ...
##  $ Product.Base.Margin: Factor w/ 61 levels "14.50%","3.60%",..: 43 25 7 31 44 34 13 10 7 34 ...
##  $ Department         : Factor w/ 3 levels "Furniture","Office Supplies",..: 2 2 2 1 1 2 2 2 2 2 ...
##  $ Container          : Factor w/ 7 levels "Jumbo Box","Jumbo Drum",..: 5 3 5 2 2 5 7 7 5 5 ...
##  $ Category           : Factor w/ 17 levels "Appliances","Binders and Binder Accessories",..: 15 15 2 4 4 15 11 11 2 15
...
##  $ Item               : Factor w/ 1263 levels "\"While you Were Out\" Message Book, One Form per Page",..: 862 921 229 60
6 625 395 276 1115 229 395 ...
##  $ Customer.Segment   : Factor w/ 4 levels "Consumer","Corporate",..: 4 3 1 2 2 2 1 1 1 2 ...
##  $ Customer_ID        : int  1656 2211 949 68 68 68 1154 1154 950 67 ...
##  $ Customer.Name      : Factor w/ 3403 levels "Aaron Davies Bruce",..: 1686 133 981 2946 2946 2946 2219 2219 1430 916 ...
##  $ Region             : Factor w/ 4 levels "AsiaPac","EMEA",..: 1 4 4 4 4 4 1 1 4 4 ...
##  $ State              : Factor w/ 149 levels "?saka","Addis Ababa",..: 22 66 19 85 85 85 1 1 71 19 ...
##  $ Country...Region   : Factor w/ 50 levels "Algeria","Argentina",..: 14 49 49 49 49 49 25 25 49 49 ...
##  $ City               : Factor w/ 1523 levels "Aberdeen","Abidjan",..: 1327 136 760 916 916 916 992 992 1100 893 ...
##  $ Ship.Date          : Factor w/ 1459 levels "01/01/2011","01/01/2012",..: 48 97 144 48 144 385 144 48 144 385 ...
##  $ Ship.Mode          : Factor w/ 3 levels "Delivery Truck",..: 3 2 3 1 1 3 2 3 3 3 ...
```

```
summary(salesData)
```

```
##        Order.Priority      Order.Date        Order          Discount
##  Critical     :3216   28/03/2013:   47   Min.   :    3   1%     :1599
##  High         :3536   15/09/2012:   40   1st Qu.:29858   5%     :1564
##  Low          :3440   05/01/2013:   37   Median :72896   3%     :1547
##  Medium       :3262   18/10/2013:   36   Mean   :59335   9%     :1543
##  Not Specified:3344   19/11/2012:   34   3rd Qu.:88699   4%     :1525
##                       21/07/2013:   34   Max.   :91591   2%     :1518
##                       (Other)   :16570                   (Other):7502
##    Unit.Price      Order.Quantity       Sales              Profit
##  Min.   :   1.00   Min.   :  1.00   Min.   :    0.90   Min.   :-4301.08
##  1st Qu.:   6.00   1st Qu.:  8.00   1st Qu.:   90.11   1st Qu.:   28.52
##  Median :  21.00   Median : 16.00   Median :  336.00   Median :  133.65
##  Mean   :  89.33   Mean   : 26.22   Mean   : 1790.07   Mean   :  882.15
##  3rd Qu.:  86.00   3rd Qu.: 38.00   3rd Qu.: 1391.16   3rd Qu.:  655.66
##  Max.   :6783.00   Max.   :288.00   Max.   :99130.12   Max.   :60250.64
##
##  Shipping.Cost    Product.Base.Margin           Department        Container
##  Min.   :  0.00   37.00% : 1474       Furniture      :3448   Jumbo Box :1064
##  1st Qu.:  3.00   38.00% : 1266       Office Supplies:9220   Jumbo Drum:1248
##  Median :  6.00   36.00% : 1190       Technology     :4130   Large Box : 812
##  Mean   : 12.86   59.00% :  962                              Medium Box: 732
##  3rd Qu.: 14.00   56.00% :  918                              Small Box :8694
##  Max.   :165.00   57.00% :  918                              Small Pack:1912
##                   (Other):10070                              Wrap Bag  :2336
##                            Category
##  Paper                         :2450
##  Binders and Binder Accessories:1830
##  Telephones and Communication  :1766
##  Office Furnishings            :1576
##  Computer Peripherals          :1516
##  Pens & Art Supplies           :1266
##  (Other)                       :6394
##                                                                  Item
##  Global High-Back Leather Tilter, Burgundy                        :   48
##  Bevis 36 x 72 Conference Tables                                  :   44
##  BoxOffice By Design Rectangular and Half-Moon Meeting Room Tables:   44
##  Fiskars® Softgrip Scissors                                       :   44
##  Master Giant Foot® Doorstop, Safety Yellow                       :   44
##  Wilson Jones Hanging View Binder, White, 1"                      :   42
##  (Other)                                                          :16532
##        Customer.Segment  Customer_ID            Customer.Name
##  Consumer      :3298    Min.   :   1   Rosemary Hedrick:   41
##  Corporate     :6152    1st Qu.: 912   Sylvia Barr     :   38
##  Home Office   :4064    Median :1778   Jason Fink      :   35
##  Small Business:3284    Mean   :1754   Courtney McBride:   33
##                         3rd Qu.:2593   Annie Rouse     :   30
##                         Max.   :3403   Kevin Erickson  :   29
##                                        (Other)         :16592
##            Region                  State                       Country...Region
##  AsiaPac      :3802   California     : 1021   United States of America:9426
##  EMEA         :1894   Texas          :  646   China                   :1257
##  Latam        :1620   Illinois       :  584   India                   : 746
##  North America:9482   New York       :  574   Brazil                  : 672
##                       Florida        :  522   Japan                   : 507
##                       Guangdong Sheng:  417   Mexico                  : 388
##                       (Other)        :13034   (Other)                 :3802
##            City            Ship.Date              Ship.Mode
##  Guangzhou   :  357   21/05/2012:   38   Delivery Truck: 2292
##  Buenos Aires:  341   09/05/2013:   35   Express Air   : 1966
##  Seoul       :  292   27/05/2013:   34   Regular Air   :12540
##  Tokyo       :  286   04/10/2013:   33
##  Paris       :  248   30/03/2013:   33
##  Beijing     :  245   02/06/2013:   31
##  (Other)     :15029   (Other)   :16594
```

```
# Numeric and factor variables
isNumericColArr <- unlist(lapply(salesData, is.numeric), use.names = FALSE)
isFactorColArr <- unlist(lapply(salesData, is.factor), use.names = FALSE)

# Numeric: "Order" "Unit.Price" "Order.Quantity" "Sales" "Profit" "Shipping.Cost" "Customer_ID"
colnames(salesData[, isNumericColArr])
```

```
## [1] "Order"          "Unit.Price"     "Order.Quantity" "Sales"
## [5] "Profit"         "Shipping.Cost"  "Customer_ID"
```

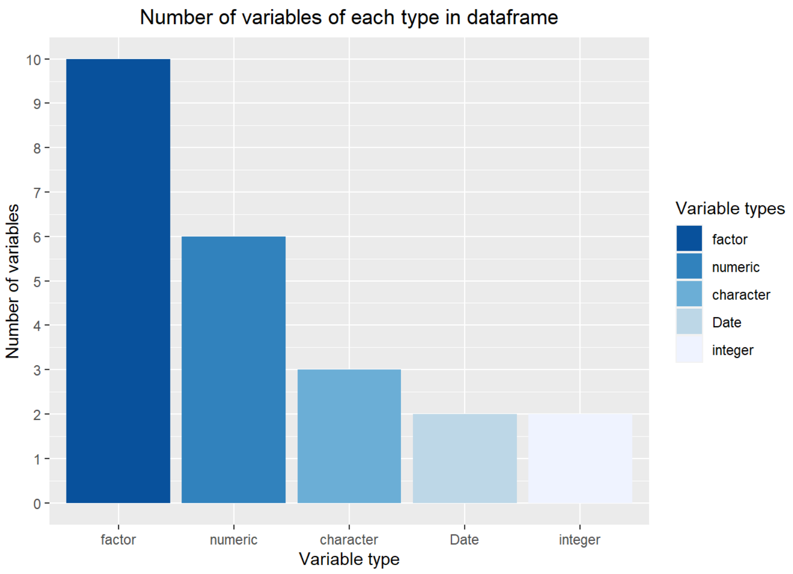
```
# Factor:
# "Order.Priority" "Order.Date" "Discount" "Product.Base.Margin" "Department" "Container"
# "Category" "Item" "Customer.Segment" "Customer.Name" "Region" "State"
# "Country...Region" "City" "Ship.Date" "Ship.Mode"
colnames(salesData[, isFactorColArr])
```

```
##  [1] "Order.Priority"      "Order.Date"          "Discount"
##  [4] "Product.Base.Margin" "Department"          "Container"
##  [7] "Category"            "Item"                "Customer.Segment"
## [10] "Customer.Name"       "Region"              "State"
## [13] "Country...Region"    "City"                "Ship.Date"
## [16] "Ship.Mode"
```

```
# Cast the data to the appropriate variable types
salesData$Order               <- as.character(salesData$Order)
salesData$Order.Date          <- as.Date(salesData$Order.Date, format = "%d/%m/%Y")
salesData$Discount            <- as.numeric(sub("%", "", salesData$Discount)) / 100
salesData$Product.Base.Margin <- as.numeric(sub("%", "", salesData$Product.Base.Margin)) / 100
salesData$Item                <- as.character(salesData$Item)
salesData$Customer.Name       <- as.character(salesData$Customer.Name)
salesData$Ship.Date           <- as.Date(salesData$Ship.Date, format = "%d/%m/%Y")
str(salesData)
```

```
## 'data.frame':    16798 obs. of  23 variables:
##  $ Order.Priority     : Factor w/ 5 levels "Critical","High",..: 2 5 1 3 3 3 1 1 1 3 ...
##  $ Order.Date         : Date, format: "2010-01-01" "2010-01-01" ...
##  $ Order              : chr  "28774" "88028" "9285" "37537" ...
##  $ Discount           : num  0.1 0.08 0.06 0 0.07 0.05 0.09 0.08 0.06 0.05 ...
##  $ Unit.Price         : num  6 96 41 292 101 155 9 15 41 155 ...
##  $ Order.Quantity     : int  32 2 3 4 43 32 16 43 1 8 ...
##  $ Sales              : num  173 177 116 1168 4039 ...
##  $ Profit             : num  106.4 45.6 33.9 605.1 2647.7 ...
##  $ Shipping.Cost      : num  5 35 3 49 45 7 2 2 3 7 ...
##  $ Product.Base.Margin: num  0.68 0.5 0.36 0.56 0.69 0.59 0.4 0.39 0.36 0.59 ...
##  $ Department         : Factor w/ 3 levels "Furniture","Office Supplies",..: 2 2 2 1 1 2 2 2 2 2 ...
##  $ Container          : Factor w/ 7 levels "Jumbo Box","Jumbo Drum",..: 5 3 5 2 2 5 7 7 5 5 ...
##  $ Category           : Factor w/ 17 levels "Appliances","Binders and Binder Accessories",..: 15 15 2 4 4 15 11 11 2 15
...
##  $ Item               : chr  "Perma STOR-ALL\231 Hanging File Box, 13 1/8\"W x 12 1/4\"D x 10 1/2\"H" "Safco Industrial W
ire Shelving" "Avery Trapezoid Ring Binder, 3\" Capacity, Black, 1040 sheets" "Hon 4070 Series Pagoda\231 Armless Upholstere
d Stacking Chairs" ...
##  $ Customer.Segment   : Factor w/ 4 levels "Consumer","Corporate",..: 4 3 1 2 2 2 1 1 1 2 ...
##  $ Customer_ID        : int  1656 2211 949 68 68 68 1154 1154 950 67 ...
##  $ Customer.Name      : chr  "Joy Corbett" "Anita Hahn" "Ernest Oh" "Scott Bunn" ...
##  $ Region             : Factor w/ 4 levels "AsiaPac","EMEA",..: 1 4 4 4 4 4 1 1 4 4 ...
##  $ State              : Factor w/ 149 levels "?saka","Addis Ababa",..: 22 66 19 85 85 85 1 1 71 19 ...
##  $ Country...Region   : Factor w/ 50 levels "Algeria","Argentina",..: 14 49 49 49 49 49 25 25 49 49 ...
##  $ City               : Factor w/ 1523 levels "Aberdeen","Abidjan",..: 1327 136 760 916 916 916 992 992 1100 893 ...
##  $ Ship.Date          : Date, format: "2010-01-02" "2010-01-03" ...
##  $ Ship.Mode          : Factor w/ 3 levels "Delivery Truck",..: 3 2 3 1 1 3 2 3 3 3 ...
```

```
ggplot(data.frame(table(sapply(salesData, class)))) +
  geom_bar(aes(x = reorder(Var1, -Freq), y = Freq,
               fill = reorder(Var1, -Freq)),
           stat = "identity") +
  scale_y_continuous(breaks = seq(0, 10, 1)) +
  labs(x = "Variable type", y = "Number of variables",
       title = "Number of variables of each type in dataframe",
       fill = "Variable types") +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_fill_brewer(palette = "Blues", direction = -1)
```



```
# Outliers and impossible data
str(salesData)
```

```
## 'data.frame':    16798 obs. of  23 variables:
##  $ Order.Priority     : Factor w/ 5 levels "Critical","High",..: 2 5 1 3 3 3 1 1 1 3 ...
##  $ Order.Date         : Date, format: "2010-01-01" "2010-01-01" ...
##  $ Order              : chr  "28774" "88028" "9285" "37537" ...
##  $ Discount           : num  0.1 0.08 0.06 0 0.07 0.05 0.09 0.08 0.06 0.05 ...
##  $ Unit.Price         : num  6 96 41 292 101 155 9 15 41 155 ...
##  $ Order.Quantity     : int  32 2 3 4 43 32 16 43 1 8 ...
##  $ Sales              : num  173 177 116 1168 4039 ...
##  $ Profit             : num  106.4 45.6 33.9 605.1 2647.7 ...
##  $ Shipping.Cost      : num  5 35 3 49 45 7 2 2 3 7 ...
##  $ Product.Base.Margin: num  0.68 0.5 0.36 0.56 0.69 0.59 0.4 0.39 0.36 0.59 ...
##  $ Department         : Factor w/ 3 levels "Furniture","Office Supplies",..: 2 2 2 1 1 2 2 2 2 2 ...
##  $ Container          : Factor w/ 7 levels "Jumbo Box","Jumbo Drum",..: 5 3 5 2 2 5 7 7 5 5 ...
##  $ Category           : Factor w/ 17 levels "Appliances","Binders and Binder Accessories",..: 15 15 2 4 4 15 11 11 2 15
...
##  $ Item               : chr  "Perma STOR-ALL\231 Hanging File Box, 13 1/8\"W x 12 1/4\"D x 10 1/2\"H" "Safco Industrial W
ire Shelving" "Avery Trapezoid Ring Binder, 3\" Capacity, Black, 1040 sheets" "Hon 4070 Series Pagoda\231 Armless Upholstere
d Stacking Chairs" ...
##  $ Customer.Segment   : Factor w/ 4 levels "Consumer","Corporate",..: 4 3 1 2 2 2 1 1 1 2 ...
##  $ Customer_ID        : int  1656 2211 949 68 68 68 1154 1154 950 67 ...
##  $ Customer.Name      : chr  "Joy Corbett" "Anita Hahn" "Ernest Oh" "Scott Bunn" ...
##  $ Region             : Factor w/ 4 levels "AsiaPac","EMEA",..: 1 4 4 4 4 4 1 1 4 4 ...
##  $ State              : Factor w/ 149 levels "?saka","Addis Ababa",..: 22 66 19 85 85 85 1 1 71 19 ...
##  $ Country...Region   : Factor w/ 50 levels "Algeria","Argentina",..: 14 49 49 49 49 49 25 25 49 49 ...
##  $ City               : Factor w/ 1523 levels "Aberdeen","Abidjan",..: 1327 136 760 916 916 916 992 992 1100 893 ...
##  $ Ship.Date          : Date, format: "2010-01-02" "2010-01-03" ...
##  $ Ship.Mode          : Factor w/ 3 levels "Delivery Truck",..: 3 2 3 1 1 3 2 3 3 3 ...
```

```
min(salesData$Discount)            # 0 >= 0      OK
```

```
## [1] 0
```

```
max(salesData$Discount)            # 0.95 <= 1   OK
```

```
## [1] 0.95
```

```
min(salesData$Unit.Price)          # 1 >= 0      OK
```

```
## [1] 1
```

```
min(salesData$Order.Quantity)      # 1 >= 1      OK
```

```
## [1] 1
```

```
min(salesData$Sales)               # 0.9 >= 0    OK
```

```
## [1] 0.9
```

```
min(salesData$Shipping.Cost)       # 0 >= 0      OK
```

```
## [1] 0
```

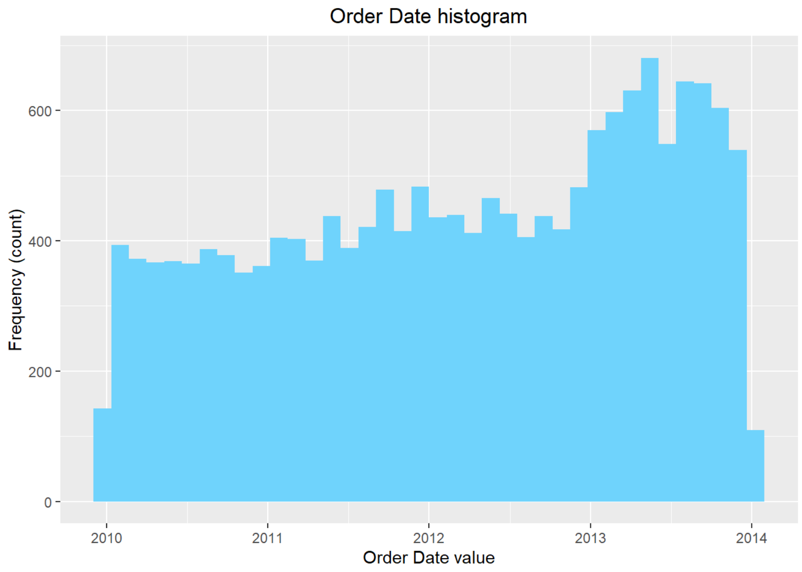
```
min(salesData$Product.Base.Margin)  # 0.036 >= 0  OK
```

```
## [1] 0.036
```

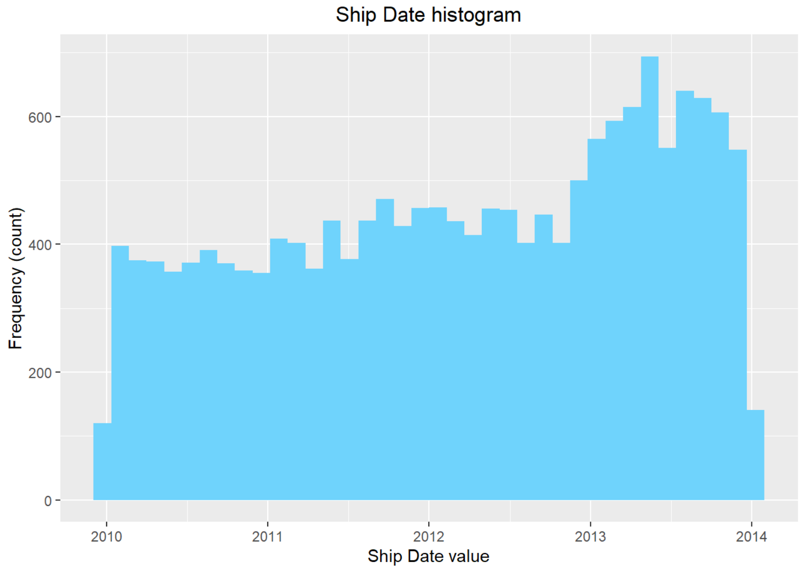
```
max(salesData$Product.Base.Margin)  # 0.85 <= 1   OK
```

```
## [1] 0.85
```

```
ggplot(salesData) +
  geom_histogram(aes(Order.Date), binwidth = 40, fill = "#6FD3FC") +
  xlab("Order Date value") + ylab("Frequency (count)") +
  ggtitle("Order Date histogram") +
  theme(plot.title = element_text(hjust = 0.5))
```



```
# Order Date values are OK.

ggplot(salesData) +
  geom_histogram(aes(Ship.Date), binwidth = 40, fill = "#6FD3FC") +
  xlab("Ship Date value") + ylab("Frequency (count)") +
  ggtitle("Ship Date histogram") +
  theme(plot.title = element_text(hjust = 0.5))
```



```
# Ship Date values are OK.

# All values are in allowed ranges.

# Remove outliers helper functions

# Detect outlier function
hasOutlier <- function(x) {
   quantile1 <- quantile(x, probs = 1/4)
   quantile3 <- quantile(x, probs = 3/4)
   IQR = quantile3 - quantile1  # Inter quartile range
   return(x > quantile3 + (IQR * 1.5) | x < quantile1 - (IQR * 1.5))
}

removeOutlier <- function(dataframe, columns = colnames(dataframe)) {
   for (col in columns) {
       # Keep observation if it doesnt have an outlier
       dataframe <- dataframe[!hasOutlier(dataframe[[col]]), ]
   }
   return(dataframe)
}

# Remove outliers
# removeOutlier(salesData, columns = c("Discount", "Unit.Price", "Order.Quantity",
#                                      "Sales", "Profit", "Shipping.Cost",
#                                      "Product.Base.Margin"))
# Outliers are not removed due to all values being real.


# saving data to CSV file
# write.csv(salesData, file = "data.csv", row.names = TRUE)



############################ PRODUCT CLASSIFICATION ############################
# install.packages(c("rpart", "rpart.plot"))
library("rpart")
library("rpart.plot")
```

```
## Warning: package 'rpart.plot' was built under R version 4.1.3
```

```
# data selection and preparation
head(salesData)
```

```
##   Order.Priority Order.Date Order Discount Unit.Price Order.Quantity   Sales
## 1           High 2010-01-01 28774     0.10          6             32  172.80
## 2  Not Specified 2010-01-01 88028     0.08         96              2  176.64
## 3       Critical 2010-01-02  9285     0.06         41              3  115.62
## 4            Low 2010-01-02 37537     0.00        292              4 1168.00
## 5            Low 2010-01-02 37537     0.07        101             43 4038.99
## 6            Low 2010-01-02 37537     0.05        155             32 4712.00
##    Profit Shipping.Cost Product.Base.Margin      Department  Container
## 1  106.36             5                0.68 Office Supplies  Small Box
## 2   45.64            35                0.50 Office Supplies  Large Box
## 3   33.90             3                0.36 Office Supplies  Small Box
## 4  605.08            49                0.56       Furniture Jumbo Drum
## 5 2647.66            45                0.69       Furniture Jumbo Drum
## 6 2671.40             7                0.59 Office Supplies  Small Box
##                         Category
## 1         Storage & Organization
## 2         Storage & Organization
## 3 Binders and Binder Accessories
## 4             Chairs & Chairmats
## 5             Chairs & Chairmats
## 6         Storage & Organization
##                                                                Item
## 1 Perma STOR-ALL\231 Hanging File Box, 13 1/8"W x 12 1/4"D x 10 1/2"H
## 2                                  Safco Industrial Wire Shelving
## 3     Avery Trapezoid Ring Binder, 3" Capacity, Black, 1040 sheets
## 4      Hon 4070 Series Pagoda\231 Armless Upholstered Stacking Chairs
## 5                                     Hon Valutask\231 Swivel Chairs
## 6                           Dual Level, Single-Width Filing Carts
##   Customer.Segment Customer_ID Customer.Name        Region      State
## 1   Small Business        1656   Joy Corbett       AsiaPac    Central
## 2      Home Office        2211    Anita Hahn North America   Maryland
## 3         Consumer         949     Ernest Oh North America California
## 4        Corporate          68    Scott Bunn North America   New York
## 5        Corporate          68    Scott Bunn North America   New York
## 6        Corporate          68    Scott Bunn North America   New York
##           Country...Region          City  Ship.Date      Ship.Mode
## 1                     Fiji          Suva 2010-01-02    Regular Air
## 2 United States of America         Bowie 2010-01-03    Express Air
## 3 United States of America   Los Angeles 2010-01-04    Regular Air
## 4 United States of America New York City 2010-01-02 Delivery Truck
## 5 United States of America New York City 2010-01-04 Delivery Truck
## 6 United States of America New York City 2010-01-09    Regular Air
```

```
str(salesData)
```

```
## 'data.frame':    16798 obs. of  23 variables:
##  $ Order.Priority     : Factor w/ 5 levels "Critical","High",..: 2 5 1 3 3 3 1 1 1 3 ...
##  $ Order.Date         : Date, format: "2010-01-01" "2010-01-01" ...
##  $ Order              : chr  "28774" "88028" "9285" "37537" ...
##  $ Discount           : num  0.1 0.08 0.06 0 0.07 0.05 0.09 0.08 0.06 0.05 ...
##  $ Unit.Price         : num  6 96 41 292 101 155 9 15 41 155 ...
##  $ Order.Quantity     : int  32 2 3 4 43 32 16 43 1 8 ...
##  $ Sales              : num  173 177 116 1168 4039 ...
##  $ Profit             : num  106.4 45.6 33.9 605.1 2647.7 ...
##  $ Shipping.Cost      : num  5 35 3 49 45 7 2 2 3 7 ...
##  $ Product.Base.Margin: num  0.68 0.5 0.36 0.56 0.69 0.59 0.4 0.39 0.36 0.59 ...
##  $ Department         : Factor w/ 3 levels "Furniture","Office Supplies",..: 2 2 2 1 1 2 2 2 2 2 ...
##  $ Container          : Factor w/ 7 levels "Jumbo Box","Jumbo Drum",..: 5 3 5 2 2 5 7 7 5 5 ...
##  $ Category           : Factor w/ 17 levels "Appliances","Binders and Binder Accessories",..: 15 15 2 4 4 15 11 11 2 15
...
##  $ Item               : chr  "Perma STOR-ALL\231 Hanging File Box, 13 1/8\"W x 12 1/4\"D x 10 1/2\"H" "Safco Industrial W
ire Shelving" "Avery Trapezoid Ring Binder, 3\" Capacity, Black, 1040 sheets" "Hon 4070 Series Pagoda\231 Armless Upholstere
d Stacking Chairs" ...
##  $ Customer.Segment   : Factor w/ 4 levels "Consumer","Corporate",..: 4 3 1 2 2 2 1 1 1 2 ...
##  $ Customer_ID        : int  1656 2211 949 68 68 68 1154 1154 950 67 ...
##  $ Customer.Name      : chr  "Joy Corbett" "Anita Hahn" "Ernest Oh" "Scott Bunn" ...
##  $ Region             : Factor w/ 4 levels "AsiaPac","EMEA",..: 1 4 4 4 4 4 1 1 4 4 ...
##  $ State              : Factor w/ 149 levels "?saka","Addis Ababa",..: 22 66 19 85 85 85 1 1 71 19 ...
##  $ Country...Region   : Factor w/ 50 levels "Algeria","Argentina",..: 14 49 49 49 49 49 25 25 49 49 ...
##  $ City               : Factor w/ 1523 levels "Aberdeen","Abidjan",..: 1327 136 760 916 916 916 992 992 1100 893 ...
##  $ Ship.Date          : Date, format: "2010-01-02" "2010-01-03" ...
##  $ Ship.Mode          : Factor w/ 3 levels "Delivery Truck",..: 3 2 3 1 1 3 2 3 3 3 ...
```

```
classificationData <- salesData[, c("Order.Priority", "Discount", "Unit.Price",
                                    "Shipping.Cost", "Department", "Category",
                                    "Customer.Segment", "Region", "Ship.Mode",
                                    "Profit")]
str(classificationData)
```

```
## 'data.frame':    16798 obs. of  10 variables:
##  $ Order.Priority  : Factor w/ 5 levels "Critical","High",..: 2 5 1 3 3 3 1 1 1 3 ...
##  $ Discount        : num  0.1 0.08 0.06 0 0.07 0.05 0.09 0.08 0.06 0.05 ...
##  $ Unit.Price      : num  6 96 41 292 101 155 9 15 41 155 ...
##  $ Shipping.Cost   : num  5 35 3 49 45 7 2 2 3 7 ...
##  $ Department      : Factor w/ 3 levels "Furniture","Office Supplies",..: 2 2 2 1 1 2 2 2 2 2 ...
##  $ Category        : Factor w/ 17 levels "Appliances","Binders and Binder Accessories",..: 15 15 2 4 4 15 11 11 2 15 ...
##  $ Customer.Segment: Factor w/ 4 levels "Consumer","Corporate",..: 4 3 1 2 2 2 1 1 1 2 ...
##  $ Region          : Factor w/ 4 levels "AsiaPac","EMEA",..: 1 4 4 4 4 4 1 1 4 4 ...
##  $ Ship.Mode       : Factor w/ 3 levels "Delivery Truck",..: 3 2 3 1 1 3 2 3 3 3 ...
##  $ Profit          : num  106.4 45.6 33.9 605.1 2647.7 ...
```

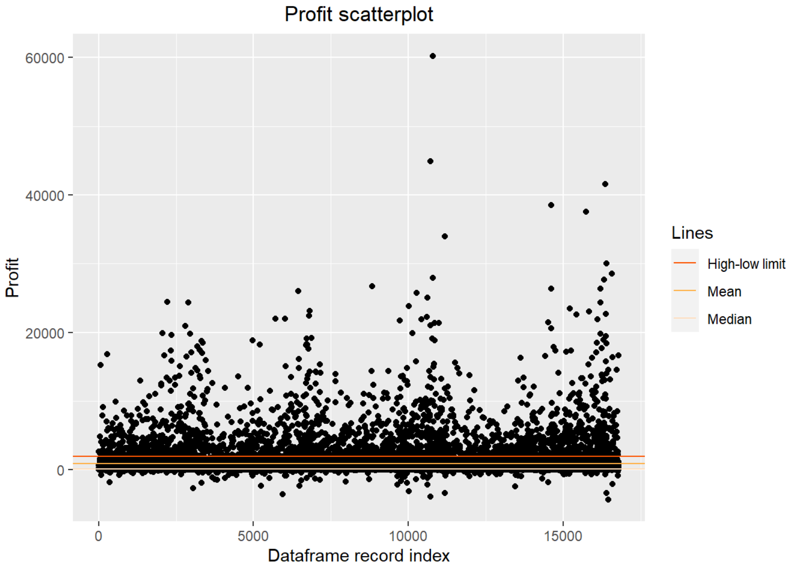
```
# Selected data includes variables which can help select products to be
# marketed. For example, it can be decided to market and advertise in specific
# regions, market and advertise specific categories of products...


# decide limit for profit (low and high)
mean(classificationData$Profit)   # 882.1462
```

```
## [1] 882.1462
```

```
median(classificationData$Profit) # 133.645
```

```
## [1] 133.645
```

```
limit <- 2000 # low - high profit limit

ggplot(classificationData) +
  geom_point(aes(x = seq_along(Profit), y = Profit)) +
  geom_hline(aes(yintercept = limit, linetype = "High-low limit"), col = "#FD5602") +
  geom_hline(aes(yintercept = mean(Profit), linetype = "Mean"), col = "#FFAF42") +
  geom_hline(aes(yintercept = median(Profit), linetype = "Median"), col = "#FEDEBE") +
  labs(x = "Dataframe record index", y = "Profit",
       title = "Profit scatterplot") +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_linetype_manual(name = "Lines", values = c(1, 1, 1),
                        guide = guide_legend(override.aes = list(color = c("#FD5602", "#FFAF42", "#FEDEBE"))))
```



```
# create new factor variable which says if profit is high or low
classificationData$ProfitFactor <- factor(ifelse(classificationData$Profit < limit, "Low", "High"))
str(classificationData)
```

```
## 'data.frame':    16798 obs. of  11 variables:
##  $ Order.Priority  : Factor w/ 5 levels "Critical","High",..: 2 5 1 3 3 3 1 1 1 3 ...
##  $ Discount        : num  0.1 0.08 0.06 0 0.07 0.05 0.09 0.08 0.06 0.05 ...
##  $ Unit.Price      : num  6 96 41 292 101 155 9 15 41 155 ...
##  $ Shipping.Cost   : num  5 35 3 49 45 7 2 2 3 7 ...
##  $ Department      : Factor w/ 3 levels "Furniture","Office Supplies",..: 2 2 2 1 1 2 2 2 2 2 ...
##  $ Category        : Factor w/ 17 levels "Appliances","Binders and Binder Accessories",..: 15 15 2 4 4 15 11 11 2 15 ...
##  $ Customer.Segment: Factor w/ 4 levels "Consumer","Corporate",..: 4 3 1 2 2 2 1 1 1 2 ...
##  $ Region          : Factor w/ 4 levels "AsiaPac","EMEA",..: 1 4 4 4 4 4 1 1 4 4 ...
##  $ Ship.Mode       : Factor w/ 3 levels "Delivery Truck",..: 3 2 3 1 1 3 2 3 3 3 ...
##  $ Profit          : num  106.4 45.6 33.9 605.1 2647.7 ...
##  $ ProfitFactor    : Factor w/ 2 levels "High","Low": 2 2 2 2 1 1 2 2 2 2 ...
```

```
head(classificationData[, c("Profit", "ProfitFactor")], 40)
```

```
##     Profit ProfitFactor
## 1   106.36          Low
## 2    45.64          Low
## 3    33.90          Low
## 4   605.08          Low
## 5  2647.66         High
## 6  2671.40         High
## 7    42.64          Low
## 8   197.95          Low
## 9     9.30          Low
## 10  662.60          Low
## 11    9.16          Low
## 12   49.15          Low
## 13   46.56          Low
## 14  116.88          Low
## 15  162.96          Low
## 16   56.16          Low
## 17  416.60          Low
## 18   10.48          Low
## 19   57.82          Low
## 20   10.50          Low
## 21    7.60          Low
## 22   24.99          Low
## 23  110.67          Low
## 24    1.40          Low
## 25  121.32          Low
## 26   65.15          Low
## 27   12.87          Low
## 28   45.04          Low
## 29  549.15          Low
## 30   -1.56          Low
## 31   10.32          Low
## 32  163.05          Low
## 33   53.52          Low
## 34   60.16          Low
## 35 1308.28          Low
## 36  147.22          Low
## 37 2350.92         High
## 38 4886.70         High
## 39  530.40          Low
## 40   42.00          Low
```

```
classificationData <- classificationData[, -grep("^Profit$", colnames(classificationData))]
str(classificationData)
```

```
## 'data.frame':    16798 obs. of  10 variables:
##  $ Order.Priority  : Factor w/ 5 levels "Critical","High",..: 2 5 1 3 3 3 1 1 1 3 ...
##  $ Discount        : num  0.1 0.08 0.06 0 0.07 0.05 0.09 0.08 0.06 0.05 ...
##  $ Unit.Price      : num  6 96 41 292 101 155 9 15 41 155 ...
##  $ Shipping.Cost   : num  5 35 3 49 45 7 2 2 3 7 ...
##  $ Department      : Factor w/ 3 levels "Furniture","Office Supplies",..: 2 2 2 1 1 2 2 2 2 2 ...
##  $ Category        : Factor w/ 17 levels "Appliances","Binders and Binder Accessories",..: 15 15 2 4 4 15 11 11 2 15 ...
##  $ Customer.Segment: Factor w/ 4 levels "Consumer","Corporate",..: 4 3 1 2 2 2 1 1 1 2 ...
##  $ Region          : Factor w/ 4 levels "AsiaPac","EMEA",..: 1 4 4 4 4 4 1 1 4 4 ...
##  $ Ship.Mode       : Factor w/ 3 levels "Delivery Truck",..: 3 2 3 1 1 3 2 3 3 3 ...
##  $ ProfitFactor    : Factor w/ 2 levels "High","Low": 2 2 2 2 1 1 2 2 2 2 ...
```

```
levels(classificationData$ProfitFactor) # ProfitFactor levels: 1 = High, 2 = Low
```

```
## [1] "High" "Low"
```

```
# divide data to train and test datasets (ratio 70:30)
RNGkind(sample.kind = "Rounding")
```

```
## Warning in RNGkind(sample.kind = "Rounding"): non-uniform 'Rounding' sampler
## used
```

```
set.seed(2)
indices <- sample(nrow(classificationData), 0.7 * nrow(classificationData))
train <- classificationData[indices, ]
test <- classificationData[-indices, ]

str(train)
```

```
## 'data.frame':    11758 obs. of  10 variables:
##  $ Order.Priority  : Factor w/ 5 levels "Critical","High",..: 5 1 5 4 2 2 3 4 1 4 ...
##  $ Discount        : num  0.07 0.07 0.04 0.06 0 0.03 0.01 0.08 0 0.08 ...
##  $ Unit.Price      : num  16 9 5 181 5 16 16 2 5 60 ...
##  $ Shipping.Cost   : num  1 6 5 26 8 1 11 2 8 4 ...
##  $ Department      : Factor w/ 3 levels "Furniture","Office Supplies",..: 2 2 2 1 2 2 3 3 2 2 ...
##  $ Category        : Factor w/ 17 levels "Appliances","Binders and Binder Accessories",..: 7 2 11 4 11 7 10 5 11 1 ...
##  $ Customer.Segment: Factor w/ 4 levels "Consumer","Corporate",..: 4 2 1 2 1 2 1 2 2 2 ...
##  $ Region          : Factor w/ 4 levels "AsiaPac","EMEA",..: 1 4 1 4 4 1 1 4 4 1 ...
##  $ Ship.Mode       : Factor w/ 3 levels "Delivery Truck",..: 3 3 3 1 3 3 3 3 3 3 ...
##  $ ProfitFactor    : Factor w/ 2 levels "High","Low": 2 2 2 2 2 2 2 2 2 2 ...
```

```
str(test)
```

```
## 'data.frame':    5040 obs. of  10 variables:
##  $ Order.Priority  : Factor w/ 5 levels "Critical","High",..: 3 3 1 3 3 2 1 2 1 1 ...
##  $ Discount        : num  0 0.05 0.09 0.05 0.04 0.05 0.01 0.09 0.07 0.03 ...
##  $ Unit.Price      : num  292 155 9 575 10 21 111 213 4 3 ...
##  $ Shipping.Cost   : num  49 7 2 24 2 21 3 52 2 6 ...
##  $ Department      : Factor w/ 3 levels "Furniture","Office Supplies",..: 1 2 2 3 2 1 3 1 2 2 ...
##  $ Category        : Factor w/ 17 levels "Appliances","Binders and Binder Accessories",..: 4 15 11 10 11 9 17 16 11 2 ...
##  $ Customer.Segment: Factor w/ 4 levels "Consumer","Corporate",..: 2 2 1 2 2 3 2 3 2 2 ...
##  $ Region          : Factor w/ 4 levels "AsiaPac","EMEA",..: 4 4 1 1 4 4 4 4 1 4 ...
##  $ Ship.Mode       : Factor w/ 3 levels "Delivery Truck",..: 1 3 2 3 3 3 3 1 2 3 ...
##  $ ProfitFactor    : Factor w/ 2 levels "High","Low": 2 1 2 2 2 2 2 2 2 2 ...
```

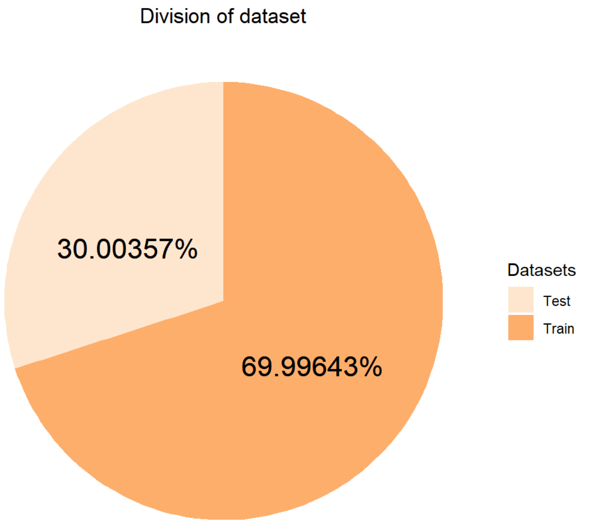
```
nrow(train) / (nrow(train) + nrow(test))  # 0.6999643
```

```
## [1] 0.6999643
```

```
nrow(test) / (nrow(train) + nrow(test))   # 0.3000357
```

```
## [1] 0.3000357
```

```
ggplot(data.frame(datasets = c("Train", "Test"),
                  percentages = c(nrow(train) / (nrow(train) + nrow(test)),
                                  nrow(test) / (nrow(train) + nrow(test)))),
       aes(x = "", y = percentages, fill = datasets)) +
  geom_bar(stat="identity", width = 1) +
  coord_polar("y", start = 0) +
  theme_void() +
  scale_fill_brewer(palette = "Oranges") +
  geom_text(aes(y = c(0.35), label = "69.99643%"), size = 6) +
  geom_text(aes(y = c(0.83), label = "30.00357%"), size = 6) +
  labs(title = "Division of dataset", fill = "Datasets") +
  theme(plot.title = element_text(hjust = 0.5))
```



Division of dataset

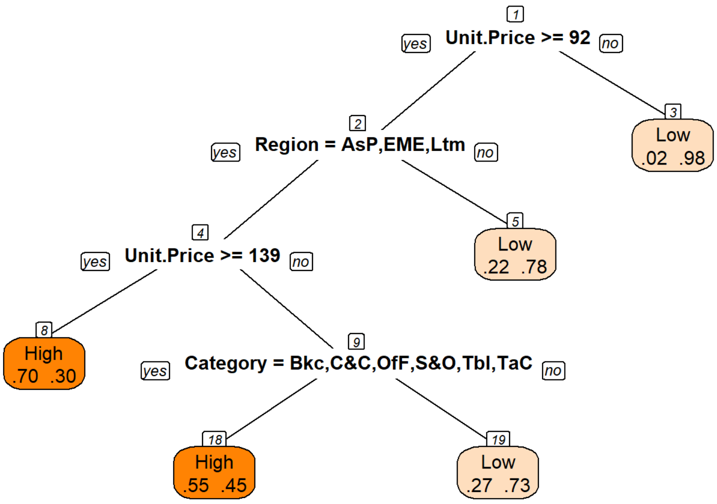
```
# Dataset is divided to 2 parts in ratio approximate to 70:30.


# create classification tree based on training data
RNGkind(sample.kind = "Rounding")
```

```
## Warning in RNGkind(sample.kind = "Rounding"): non-uniform 'Rounding' sampler
## used
```

```
set.seed(2)
tree <- rpart(ProfitFactor~., data = train, method = "class")
print(tree)
```

```
## n= 11758
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 11758 1335 Low (0.11353972 0.88646028)
##    2) Unit.Price>=91.5 2872 1157 Low (0.40285515 0.59714485)
##      4) Region=AsiaPac,EMEA,Latam 1274  467 High (0.63343799 0.36656201)
##        8) Unit.Price>=138.5 870  262 High (0.69885057 0.30114943) *
##        9) Unit.Price< 138.5 404  199 Low (0.49257426 0.50742574)
##         18) Category=Bookcases,Chairs & Chairmats,Office Furnishings,Storage & Organization,Tables,Telephones and Communi
cation 316  141 High (0.55379747 0.44620253) *
##         19) Category=Appliances,Binders and Binder Accessories,Computer Peripherals,Labels,Office Machines,Paper 88   24
Low (0.27272727 0.72727273) *
##      5) Region=North America 1598  350 Low (0.21902378 0.78097622) *
##    3) Unit.Price< 91.5 8886  178 Low (0.02003151 0.97996849) *
```

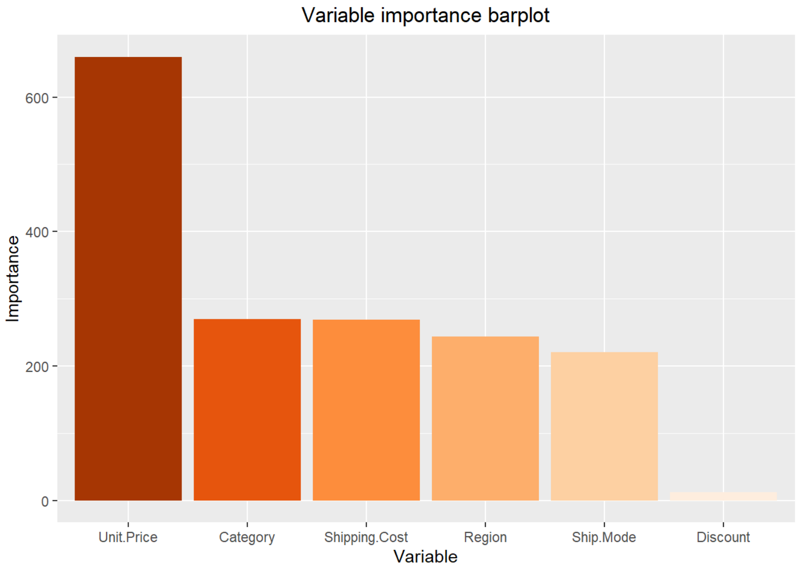
```
prp(tree, extra = 4, nn = TRUE, yesno = 2, varlen = 0,
    box.col = ifelse(tree$frame$yval == 1, "#FF8303", "#FEDEBE"))
```



```
# Tree shows classification of data. Using the tree, profit levels (high / low)
# can be predicted based on variable values of the records. Furthermore,
# expectations of profit from different products can be read from the tree.
# Consequently, decisions can be made for which products will be marketed /
# advertised.


# variable importance
print(tree$variable.importance)
```

```
##    Unit.Price      Category Shipping.Cost        Region     Ship.Mode 
##     659.85605     269.87634     269.14029     243.47874     220.18450 
##      Discount 
##      11.96173
```

```
ggplot(data = data.frame(tree$variable.importance,
                         variable = names(tree$variable.importance))) +
  geom_bar(aes(x = reorder(variable, -tree.variable.importance),
               y = tree.variable.importance,
               fill = reorder(variable, tree.variable.importance)),
           stat = "identity") +
  scale_fill_brewer(palette = "Oranges") +
  theme(legend.position = "none") +
  labs(x = "Variable", y = "Importance",
       title = "Variable importance barplot") +
  theme(plot.title = element_text(hjust = 0.5))
```



```
# Prices of products are the most important, which is expected. Product
# categories and regions are important variables which can be used. Also,
# interesting result is low importance of discount.


# test classification tree on testing / validation data
prediction <- predict(tree, newdata = test, type = "class")
head(prediction)
```

```
##    4    6    7   14   21   22 
##  Low  Low  Low High  Low  Low 
## Levels: High Low
```

```
# calculate accuracy of the tree model
accuracy <- sum(prediction == test$ProfitFactor) / nrow(test) * 100
print(accuracy) # 92.5
```

```
## [1] 92.5
```

```
# Accuracy of the model is reasonably high. That means model is rather reliable
# and can be used for marketing purposes.


# pruning tree and performance advantages
print(tree$cptable)
```

```
##           CP nsplit rel error    xerror       xstd
## 1 0.12734082      0 1.0000000 1.0000000 0.02576849
## 2 0.01498127      2 0.7453184 0.7453184 0.02260634
## 3 0.01000000      4 0.7153558 0.7393258 0.02252364
```

```
#           CP nsplit rel error    xerror       xstd
# 1 0.12734082      0 1.0000000 1.0000000 0.02576849
# 2 0.01498127      2 0.7453184 0.7453184 0.02260634
# 3 0.01000000      4 0.7153558 0.7393258 0.02252364
min(tree$cptable[, "xerror"]) # 0.7393258
```

```
## [1] 0.7393258
```

```
# xstd = 0.02252364 (from cptable)
0.7393258 - 0.02252364  # 0.7168022
```

```
## [1] 0.7168022
```

```
0.7393258 + 0.02252364  # 0.7618494
```

```
## [1] 0.7618494
```

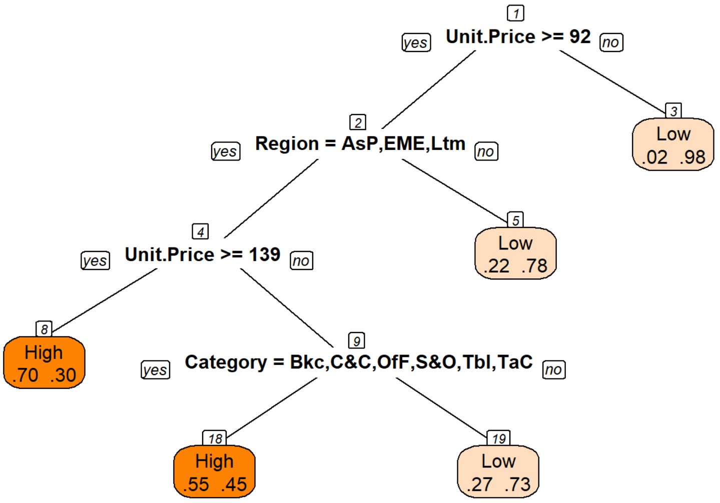
```
# interval between 0.7168022 and 0.7618494
# Only xerror values from the cptable in the interval are the ones with 2 and 4
# splits (value of nsplit variable).
min(2, 4)  # nsplit = 2
```

```
## [1] 2
```

```
# CP = 0.01498127 (from table)

prunedTree <- prune(tree, cp = 0.01498127)
print(prunedTree)
```

```
## n= 11758
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 11758 1335 Low (0.11353972 0.88646028)
##    2) Unit.Price>=91.5 2872 1157 Low (0.40285515 0.59714485)
##      4) Region=AsiaPac,EMEA,Latam 1274  467 High (0.63343799 0.36656201)
##        8) Unit.Price>=138.5 870  262 High (0.69885057 0.30114943) *
##        9) Unit.Price< 138.5 404  199 Low (0.49257426 0.50742574)
##         18) Category=Bookcases,Chairs & Chairmats,Office Furnishings,Storage & Organization,Tables,Telephones and Communi
cation 316  141 High (0.55379747 0.44620253) *
##         19) Category=Appliances,Binders and Binder Accessories,Computer Peripherals,Labels,Office Machines,Paper 88   24
Low (0.27272727 0.72727273) *
##      5) Region=North America 1598  350 Low (0.21902378 0.78097622) *
##    3) Unit.Price< 91.5 8886  178 Low (0.02003151 0.97996849) *
```

```
prp(prunedTree, extra = 4, nn = TRUE, yesno = 2, varlen = 0,
    box.col = ifelse(prunedTree$frame$yval == 1, "#FF8303", "#FEDEBE"))
```



```
# The tree is the same. There cannot be any performance advantages.



############################# CLUSTERING CUSTOMERS #############################
# install.packages(c("NbClust", "factoextra"))
library("NbClust")
```

```
## Warning: package 'NbClust' was built under R version 4.1.3
```

```
library("factoextra")
```

```
## Warning: package 'factoextra' was built under R version 4.1.3
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
# data selection and preparation
head(salesData)
```

```
##   Order.Priority Order.Date Order Discount Unit.Price Order.Quantity   Sales
## 1           High 2010-01-01 28774     0.10          6             32  172.80
## 2  Not Specified 2010-01-01 88028     0.08         96              2  176.64
## 3       Critical 2010-01-02  9285     0.06         41              3  115.62
## 4            Low 2010-01-02 37537     0.00        292              4 1168.00
## 5            Low 2010-01-02 37537     0.07        101             43 4038.99
## 6            Low 2010-01-02 37537     0.05        155             32 4712.00
##    Profit Shipping.Cost Product.Base.Margin      Department  Container
## 1  106.36             5                0.68 Office Supplies  Small Box
## 2   45.64            35                0.50 Office Supplies  Large Box
## 3   33.90             3                0.36 Office Supplies  Small Box
## 4  605.08            49                0.56       Furniture Jumbo Drum
## 5 2647.66            45                0.69       Furniture Jumbo Drum
## 6 2671.40             7                0.59 Office Supplies  Small Box
##                         Category
## 1         Storage & Organization
## 2         Storage & Organization
## 3 Binders and Binder Accessories
## 4             Chairs & Chairmats
## 5             Chairs & Chairmats
## 6         Storage & Organization
##                                                                Item
## 1 Perma STOR-ALL\231 Hanging File Box, 13 1/8"W x 12 1/4"D x 10 1/2"H
## 2                                   Safco Industrial Wire Shelving
## 3     Avery Trapezoid Ring Binder, 3" Capacity, Black, 1040 sheets
## 4      Hon 4070 Series Pagoda\231 Armless Upholstered Stacking Chairs
## 5                                      Hon Valutask\231 Swivel Chairs
## 6                            Dual Level, Single-Width Filing Carts
##   Customer.Segment Customer_ID Customer.Name        Region      State
## 1   Small Business        1656   Joy Corbett       AsiaPac    Central
## 2      Home Office        2211    Anita Hahn North America   Maryland
## 3         Consumer         949     Ernest Oh North America California
## 4        Corporate          68    Scott Bunn North America   New York
## 5        Corporate          68    Scott Bunn North America   New York
## 6        Corporate          68    Scott Bunn North America   New York
##           Country...Region          City  Ship.Date      Ship.Mode
## 1                     Fiji          Suva 2010-01-02    Regular Air
## 2 United States of America         Bowie 2010-01-03    Express Air
## 3 United States of America   Los Angeles 2010-01-04    Regular Air
## 4 United States of America New York City 2010-01-02 Delivery Truck
## 5 United States of America New York City 2010-01-04 Delivery Truck
## 6 United States of America New York City 2010-01-09    Regular Air
```

```
str(salesData)
```

```
## 'data.frame':    16798 obs. of  23 variables:
##  $ Order.Priority     : Factor w/ 5 levels "Critical","High",..: 2 5 1 3 3 3 1 1 1 3 ...
##  $ Order.Date         : Date, format: "2010-01-01" "2010-01-01" ...
##  $ Order              : chr  "28774" "88028" "9285" "37537" ...
##  $ Discount           : num  0.1 0.08 0.06 0 0.07 0.05 0.09 0.08 0.06 0.05 ...
##  $ Unit.Price         : num  6 96 41 292 101 155 9 15 41 155 ...
##  $ Order.Quantity     : int  32 2 3 4 43 32 16 43 1 8 ...
##  $ Sales              : num  173 177 116 1168 4039 ...
##  $ Profit             : num  106.4 45.6 33.9 605.1 2647.7 ...
##  $ Shipping.Cost      : num  5 35 3 49 45 7 2 2 3 7 ...
##  $ Product.Base.Margin: num  0.68 0.5 0.36 0.56 0.69 0.59 0.4 0.39 0.36 0.59 ...
##  $ Department         : Factor w/ 3 levels "Furniture","Office Supplies",..: 2 2 2 1 1 2 2 2 2 2 ...
##  $ Container          : Factor w/ 7 levels "Jumbo Box","Jumbo Drum",..: 5 3 5 2 2 5 7 7 5 5 ...
##  $ Category           : Factor w/ 17 levels "Appliances","Binders and Binder Accessories",..: 15 15 2 4 4 15 11 11 2 15
...
##  $ Item               : chr  "Perma STOR-ALL\231 Hanging File Box, 13 1/8\"W x 12 1/4\"D x 10 1/2\"H" "Safco Industrial W
ire Shelving" "Avery Trapezoid Ring Binder, 3\" Capacity, Black, 1040 sheets" "Hon 4070 Series Pagoda\231 Armless Upholstere
d Stacking Chairs" ...
##  $ Customer.Segment   : Factor w/ 4 levels "Consumer","Corporate",..: 4 3 1 2 2 2 1 1 1 2 ...
##  $ Customer_ID        : int  1656 2211 949 68 68 68 1154 1154 950 67 ...
##  $ Customer.Name      : chr  "Joy Corbett" "Anita Hahn" "Ernest Oh" "Scott Bunn" ...
##  $ Region             : Factor w/ 4 levels "AsiaPac","EMEA",..: 1 4 4 4 4 4 1 1 4 4 ...
##  $ State              : Factor w/ 149 levels "?saka","Addis Ababa",..: 22 66 19 85 85 85 1 1 71 19 ...
##  $ Country...Region   : Factor w/ 50 levels "Algeria","Argentina",..: 14 49 49 49 49 49 25 25 49 49 ...
##  $ City               : Factor w/ 1523 levels "Aberdeen","Abidjan",..: 1327 136 760 916 916 916 992 992 1100 893 ...
##  $ Ship.Date          : Date, format: "2010-01-02" "2010-01-03" ...
##  $ Ship.Mode          : Factor w/ 3 levels "Delivery Truck",..: 3 2 3 1 1 3 2 3 3 3 ...
```

```
# select data needed for clustering
clusteringData <- salesData[, c("Discount", "Unit.Price", "Order.Quantity",
                                "Department", "Customer_ID")]
str(clusteringData)
```

```
## 'data.frame':    16798 obs. of  5 variables:
##  $ Discount      : num  0.1 0.08 0.06 0 0.07 0.05 0.09 0.08 0.06 0.05 ...
##  $ Unit.Price    : num  6 96 41 292 101 155 9 15 41 155 ...
##  $ Order.Quantity: int  32 2 3 4 43 32 16 43 1 8 ...
##  $ Department    : Factor w/ 3 levels "Furniture","Office Supplies",..: 2 2 2 1 1 2 2 2 2 2 ...
##  $ Customer_ID   : int  1656 2211 949 68 68 68 1154 1154 950 67 ...
```

```
# Discount, Unit.Price and Order.Quantity variables will be used to calculate
# total money spent by each customer. Customers will be represented by its IDs.
# Total money spent by each customer will be calculated for each product
# department (Technology, Office.Supplies, Furniture).


# aggragate and change data to desired shape
clusteringData$TotalSpent <- clusteringData$Order.Quantity * clusteringData$Unit.Price * (1 - clusteringData$Discount)
clusteringData$Order.Quantity <- NULL
clusteringData$Unit.Price     <- NULL
clusteringData$Discount       <- NULL
clusteringData <- pivot_wider(clusteringData, names_from = Department, values_from = TotalSpent, values_fn = sum, values_fil
l = 0)
clusteringData <- data.frame(clusteringData)
head(clusteringData)
```

```
##   Customer_ID Office.Supplies Furniture Technology
## 1        1656          172.80      0.00       0.00
## 2        2211          341.38      0.00       0.00
## 3         949         3416.78   3478.08   14392.21
## 4          68        25020.53  10691.56   19033.74
## 5        1154          724.44    530.88    8808.24
## 6         950          277.62      0.00    2223.95
```

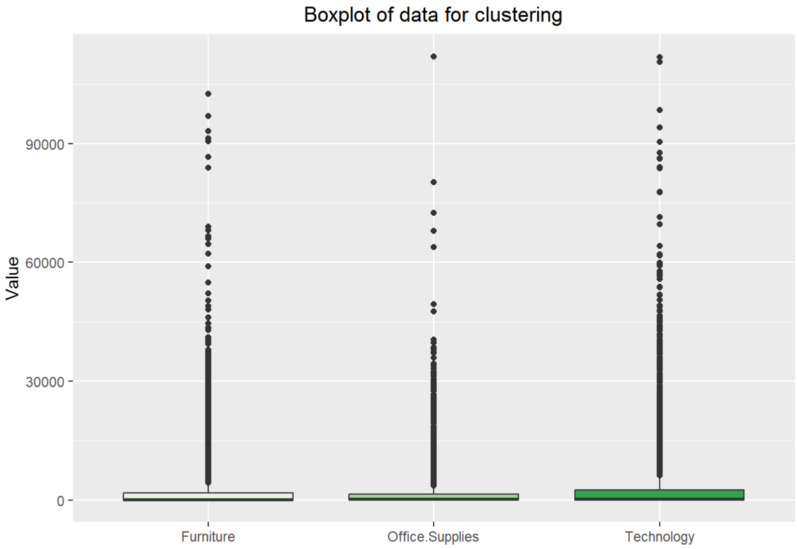
```
nrow(clusteringData)  # 3403
```

```
## [1] 3403
```

```
str(clusteringData)
```

```
## 'data.frame':    3403 obs. of  4 variables:
##  $ Customer_ID    : int  1656 2211 949 68 1154 950 67 1155 117 168 ...
##  $ Office.Supplies: num  173 341 3417 25021 724 ...
##  $ Furniture      : num  0 0 3478 10692 531 ...
##  $ Technology     : num  0 0 14392 19034 8808 ...
```

```
ggplot(clusteringData) +
  geom_boxplot(aes(y = Furniture, x = "Furniture",
                   fill = "Furniture")) +
  geom_boxplot(aes(y = Office.Supplies, x = "Office.Supplies",
                   fill = "Office.Supplies")) +
  geom_boxplot(aes(y = Technology, x = "Technology",
                   fill = "Technology")) +
  labs(x = "", y = "Value",
       title = "Boxplot of data for clustering") +
  theme(plot.title = element_text(hjust = 0.5),
        legend.position="none") +
  scale_fill_brewer(palette = "Greens")
```



```
# Total spending values are grouped by department instead of by category to
# avoid many variables and very large amount of zeroes.


# standardize data
colnames(clusteringData)[1] # "Customer_ID"
```

```
## [1] "Customer_ID"
```

```
clusteringDataScaled <- scale(clusteringData[, -1])
head(clusteringDataScaled)
```

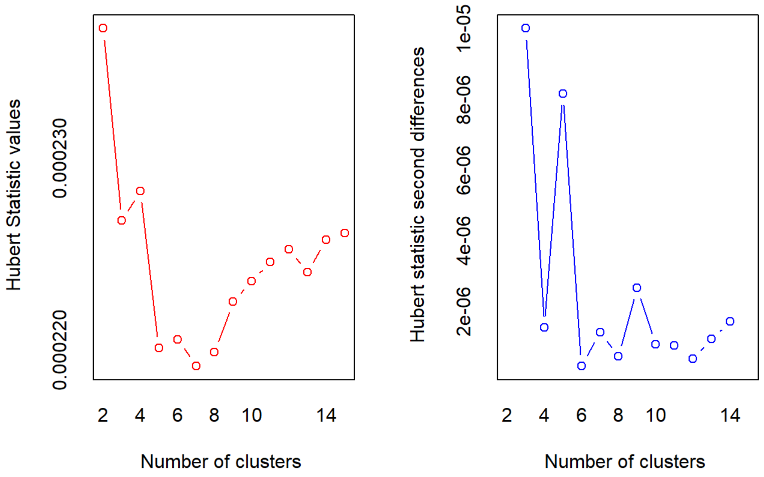
```
##      Office.Supplies   Furniture Technology
## [1,]      -0.3460002 -0.36001184 -0.3833023
## [2,]      -0.3173320 -0.36001184 -0.3833023
## [3,]       0.2056618  0.06448951  1.1161356
## [4,]       3.8795338  0.94489824  1.5997087
## [5,]      -0.2521899 -0.29521768  0.5343753
## [6,]      -0.3281748 -0.36001184 -0.1516024
```

```
str(clusteringDataScaled)
```

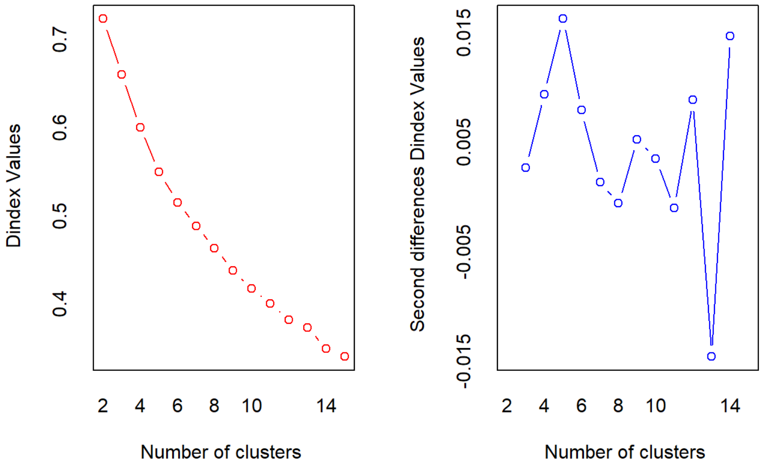
```
##  num [1:3403, 1:3] -0.346 -0.317 0.206 3.88 -0.252 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr [1:3] "Office.Supplies" "Furniture" "Technology"
##  - attr(*, "scaled:center")= Named num [1:3] 2207 2950 3679
##   ..- attr(*, "names")= chr [1:3] "Office.Supplies" "Furniture" "Technology"
##  - attr(*, "scaled:scale")= Named num [1:3] 5880 8193 9598
##   ..- attr(*, "names")= chr [1:3] "Office.Supplies" "Furniture" "Technology"
```

```
# Customer_ID variable is left out of standardized data because it is not
# important for finding number of clusters and grouping data.


# Partitional clustering is selected to enable iterative relocation. Data will
# be clustered using Kmeans method because it is good for large amounts of data.
# Euclidean distance will be used due to regular distance between two values
# being important.

# find optimal number of clusters
# (partitional clustering, euclidean distance, kmeans method)
numberOfClusters <- NbClust(clusteringDataScaled,
                            distance = "euclidean", method = "kmeans",
                            min.nc = 2, max.nc = 15)
```

```
## [1] "Frey index : No clustering structure in this data set"
```



```
## *** : The Hubert index is a graphical method of determining the number of clusters.
##                 In the plot of Hubert index, we seek a significant knee that corresponds to a
##                 significant increase of the value of the measure i.e the significant peak in Hubert
##                 index second differences plot.
##
```



```
## *** : The D index is a graphical method of determining the number of clusters.
##                 In the plot of D index, we seek a significant knee (the significant peak in Dindex
##                 second differences plot) that corresponds to a significant increase of the value of
##                 the measure.
##
## *******************************************************************
## * Among all indices:
## * 8 proposed 2 as the best number of clusters
## * 2 proposed 3 as the best number of clusters
## * 3 proposed 5 as the best number of clusters
## * 1 proposed 7 as the best number of clusters
## * 2 proposed 11 as the best number of clusters
## * 4 proposed 12 as the best number of clusters
## * 1 proposed 13 as the best number of clusters
## * 1 proposed 14 as the best number of clusters
## * 1 proposed 15 as the best number of clusters
##
##                    ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  2
##
##
## *******************************************************************
```

```
print(numberOfClusters)
```

```
## $All.index
##        KL       CH Hartigan      CCC     Scott     Marriot    TrCovW   TraceW
## 2  0.5600 2242.942 721.2604  -7.8605  3904.944 33439366477 4598812.8 6150.064
## 3  2.1283 1719.429 869.2853 -17.2820  6385.653 36295255260 3254701.1 5074.005
## 4  0.5223 1728.613 841.8331 -15.8488  8753.012 32181215372 2362401.0 4040.867
## 5  1.7658 1827.475 609.8716  -6.7112 10492.204 30162327843 1010883.7 3238.729
## 6  1.6363 1845.807 472.8834  -1.9163 12356.623 25112400107  808315.2 2745.897
## 7  2.9218 1830.570 352.0312   0.8257 13614.681 23617128408  529269.9 2410.360
## 8  0.9640 1781.475 240.9011   1.6616 14680.759 22550545485  449812.1 2183.968
## 9  0.9216 1699.011 286.5834   0.6320 15366.931 23328779465  424434.8 2039.267
## 10 0.6352 1669.104 161.7029   1.5535 16161.784 22801688158  377400.3 1880.482
## 11 0.1531 1589.486 646.9608  -0.0556 16749.255 23215539229  356522.2 1794.939
## 12 4.1366 1778.882 296.0538  10.2271 18314.356 17442793667  238259.2 1507.426
## 13 1.9382 1797.148 115.9663  12.7045 19234.924 15619112877  193940.1 1386.386
## 14 0.8371 1724.065 227.1049  11.1486 19457.810 16966052542  173411.7 1340.529
## 15 1.2117 1723.912  94.2155  12.6096 20066.717 16285376382  164502.3 1256.339
##    Friedman  Rubin Cindex     DB Silhouette   Duda   Pseudot2   Beale Ratkowsky
## 2    2.0210 1.6595 0.0309 1.2072     0.7849 2.2773 -1843.0610 -0.9545    0.4457
## 3    3.6290 2.0114 0.0276 1.5750     0.7617 2.0758 -1689.5023 -0.8820    0.4072
## 4    5.2107 2.5257 0.0280 1.2565     0.7393 1.0611   -25.3417 -0.0977    0.3884
## 5    6.6455 3.1512 0.0252 1.0822     0.7129 0.8070   709.7735  0.4062    0.3695
## 6    8.7312 3.7168 0.0236 1.0355     0.6998 1.4570  -106.9583 -0.5303    0.3490
## 7   10.1885 4.2342 0.0256 0.9687     0.6919 1.5347   -44.2477 -0.5885    0.3303
## 8   11.7761 4.6731 0.0234 0.9831     0.6594 1.7270   -64.4068 -0.7090    0.3134
## 9   12.7698 5.0047 0.0191 1.1219     0.6076 2.4413  -138.7389 -0.9973    0.2981
## 10  14.0537 5.4273 0.0170 1.1405     0.5922 1.8084   -46.0420 -0.7546    0.2856
## 11  15.2042 5.6860 0.0156 1.1659     0.5931 2.0498  -119.3323 -0.8628    0.2737
## 12  17.8997 6.7705 0.0235 1.0251     0.5856 1.1638   -31.8122 -0.2373    0.2665
## 13  19.9828 7.3616 0.0232 0.9827     0.5883 1.3514   -15.0831 -0.4280    0.2578
## 14  20.2437 7.6134 0.0205 1.0005     0.5984 1.8701   -47.9220 -0.7774    0.2491
## 15  21.5941 8.1236 0.0196 1.0255     0.5928 1.1080   -28.6532 -0.1632    0.2418
##         Ball Ptbiserial   Frey McClain   Dunn Hubert SDindex Dindex   SDbw
## 2  3075.0322     0.7721 4.5904  0.0270 0.0045  2e-04 10.5894 0.7228 2.7679
## 3  1691.3349     0.7716 4.8775  0.0329 0.0049  2e-04 11.5814 0.6601 3.2034
## 4  1010.2169     0.7566 4.8065  0.0397 0.0049  2e-04 13.9206 0.6000 3.6407
## 5   647.7458     0.7356 4.4513  0.0462 0.0063  2e-04 10.3869 0.5493 2.9412
## 6   457.6494     0.7264 5.6768  0.0487 0.0029  2e-04 10.8109 0.5149 2.7059
## 7   344.3371     0.7221 6.1135  0.0497 0.0041  2e-04 10.4762 0.4884 2.5309
## 8   272.9960     0.6988 6.0611  0.0552 0.0015  2e-04  9.7751 0.4632 2.2746
## 9   226.5852     0.6392 5.6063  0.0699 0.0012  2e-04 10.6183 0.4375 2.1842
## 10  188.0482     0.6094 4.9790  0.0774 0.0011  2e-04 10.4061 0.4170 2.0929
## 11  163.1763     0.5906 3.2115  0.0818 0.0010  2e-04 10.3523 0.4000 2.0261
## 12  125.6188     0.5991 3.0029  0.0809 0.0012  2e-04  9.7540 0.3820 1.7305
## 13  106.6451     0.5977 4.0472  0.0811 0.0018  2e-04  9.4412 0.3729 1.6443
## 14   95.7521     0.5675 6.0284  0.0862 0.0019  2e-04  9.1239 0.3492 1.5542
## 15   83.7559     0.5556 7.3208  0.0893 0.0011  2e-04  9.2934 0.3402 1.5263
## 
## $All.CriticalValues
##    CritValue_Duda CritValue_PseudoT2 Fvalue_Beale
## 2          0.7477          1108.7866       1.0000
## 3          0.7473          1102.6610       1.0000
## 4          0.6628           223.8292       1.0000
## 5          0.6718          1449.8006       0.7486
## 6          0.6024           225.0414       1.0000
## 7          0.5905            88.0706       1.0000
## 8          0.5588           120.8073       1.0000
## 9          0.5928           161.4399       1.0000
## 10         0.5840            73.3597       1.0000
## 11         0.5612           182.2112       1.0000
## 12         0.5689           171.2438       1.0000
## 13         0.3869            91.9094       1.0000
## 14         0.4868           108.5771       1.0000
## 15         0.5043           288.9641       1.0000
## 
## $Best.nc
##                      KL       CH Hartigan     CCC    Scott    Marriot  TrCovW
## Number_clusters 12.0000    2.000  11.0000 13.0000    3.000         12       5
## Value_Index      4.1366 2242.942 485.2579 12.7045 2480.709 3949064771 1351517
##                   TraceW Friedman   Rubin  Cindex     DB Silhouette   Duda
## Number_clusters   5.0000  12.0000 12.0000 11.0000 7.0000     2.0000 2.0000
## Value_Index     309.3063   2.6955 -0.4934  0.0156 0.9687     0.7849 2.2773
##                  PseudoT2   Beale Ratkowsky     Ball PtBiserial Frey McClain
## Number_clusters     2.000  2.0000    2.0000    3.000     2.0000   NA   2.000
## Value_Index     -1843.061 -0.9545    0.4457 1383.697     0.7721   NA   0.027
##                   Dunn Hubert SDindex Dindex    SDbw
## Number_clusters 5.0000      0 14.0000      0 15.0000
## Value_Index     0.0063      0  9.1239      0  1.5263
## 
## $Best.partition
##    [1] 1 1 1 2 1 1 1 1 1 2 1 2 1 1 1 1 1 1 1 2 1 1 1 1 1 2 1 2 1 1 1 1 1 1 1 1 1
##   [38] 2 2 2 1 1 1 1 1 1 1 1 1 1 2 1 2 2 1 1 1 1 1 2 1 1 1 1 2 1 2 1 1 1 1 1 1 1
##   [75] 1 1 2 2 2 2 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 2 1 1 1 1 2
##  [112] 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1
##  [149] 1 1 1 1 1 2 1 1 1 1 2 1 1 1 1 1 1 2 1 2 2 1 2 1 1 1 1 1 2 2 1 2 2 2 1 1 1
##  [186] 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 2 2 2 1 1 1 1
##  [223] 1 1 2 1 1 1 1 2 2 1 1 1 1 1 2 2 1 1 2 1 1 1 1 1 1 2 1 1 1 2 1 1 1 1 1 1 1
##  [260] 1 1 2 2 1 1 1 1 1 1 1 2 2 2 1 1 1 1 2 1 1 1 1 2 1 1 2 1 1 1 1 1 2 1 1 1 1
##  [297] 2 1 2 1 2 1 1 1 1 1 1 1 2 1 1 1 1 2 2 2 1 1 1 1 1 1 1 1 1 1 2 1 1 2 1 1 1
##  [334] 1 2 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 1 1 1 1 1 2 1 1 1 1 2 1 1 1 2 2 2 1 1
##  [371] 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 2 1 1 1 1 1 2 1 1 1 1 1 1 2 1 1 2 1 1
##  [408] 1 2 2 2 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 1 1 1 1 2 1 1 1 1 2
##  [445] 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1
##  [482] 1 1 1 1 1 1 1 1 2 1 1 1 1 2 2 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 2 1 1 1 1 1
##  [519] 1 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1
##  [556] 1 1 1 1 1 1 1 1 2 1 1 1 1 1 2 1 1 1 2 2 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1
##  [593] 1 1 1 1 1 2 1 1 1 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 2 1 1
##  [630] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 2 1 1 1 1 2 1 1 1 1 1 1 1 1 1 2 1 1
##  [667] 1 1 1 1 1 1 1 1 2 2 1 1 2 1 1 1 1 1 1 1 2 2 2 1 1 1 2 2 1 1 1 1 2 2 1 1 1
##  [704] 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 2 1 1 1 1 1 1 2 1 1 1 1 1 2 1 1
##  [741] 1 1 2 1 2 1 1 2 2 1 1 1 1 2 2 1 1 1 1 2 1 1 2 2 2 1 1 1 1 1 2 2 1 1 1 2 1
##  [778] 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 2
##  [815] 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 2 1 2 1 1 1 1 2
##  [852] 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1
##  [889] 1 1 1 1 1 1 1 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 2 1 2 1 1 1 1 1 2
##  [926] 1 1 1 1 1 2 1 1 2 1 1 1 1 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 2
##  [963] 1 1 1 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 2 1 2 1 1 2 1
## [1000] 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 2 1 1 1 2 1 1 1 2 2 1 1
## [1037] 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1
## [1074] 1 1 1 1 1 2 2 1 1 1 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1
## [1111] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2
## [1148] 1 2 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1185] 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1
## [1222] 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 2 1 1 1
## [1259] 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1296] 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1333] 2 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 2 1 1 2 1 1 1 1 2 1 1 1 1 2 1
## [1370] 1 1 2 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 2 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1
## [1407] 1 1 1 1 1 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 1 1 1 1 1 2 1 1 1 1 1
## [1444] 1 2 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1
## [1481] 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1518] 1 2 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 2 1
## [1555] 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2
## [1592] 1 1 1 1 2 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1
## [1629] 1 1 2 2 1 1 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 1 1 1 2 1 1 1 1 1 2
## [1666] 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1703] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1
## [1740] 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1
## [1777] 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1814] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1851] 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1888] 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 2 1
## [1925] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1962] 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1999] 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [2036] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 2 1 1
## [2073] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1
## [2110] 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [2147] 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [2184] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1
## [2221] 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 2 1 1 1 1 2
## [2258] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [2295] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1
## [2332] 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [2369] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1
## [2406] 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [2443] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2
## [2480] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [2517] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [2554] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [2591] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1
## [2628] 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [2665] 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [2702] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [2739] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [2776] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1
## [2813] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [2850] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [2887] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [2924] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1
## [2961] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [2998] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [3035] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [3072] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [3109] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [3146] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [3183] 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [3220] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [3257] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [3294] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [3331] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [3368] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

```
# The optimal number of clusters is 2.

# clustering votes
table(numberOfClusters$Best.nc[1,]) # 2
```

```
## 
##  0  2  3  5  7 11 12 13 14 15 
##  2  8  2  3  1  2  4  1  1  1
```

```
# Clustering votes also show, with significant difference, that the optimal
# number of clusters is 2.

# group data
RNGkind(sample.kind = "Rounding")
```

```
## Warning in RNGkind(sample.kind = "Rounding"): non-uniform 'Rounding' sampler
## used
```

```
set.seed(2)
groups <- kmeans(clusteringDataScaled, 2, nstart = 25)
print(groups)
```

```
## K-means clustering with 2 clusters of sizes 303, 3100
## 
## Cluster means:
##   Office.Supplies  Furniture Technology
## 1       1.9593805  2.0270059  2.0606141
## 2      -0.1915136 -0.1981235 -0.2014084
## 
## Clustering vector:
##    [1] 2 2 2 1 2 2 2 2 2 1 2 1 2 2 2 2 2 2 2 1 2 2 2 2 2 1 2 1 2 2 2 2 2 2 2 2 2
##   [38] 1 1 1 2 2 2 2 2 2 2 2 2 2 1 2 1 1 2 2 2 2 2 1 2 2 2 2 1 2 1 2 2 2 2 2 2 2
##   [75] 2 2 1 1 1 1 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 1 2 2 2 2 1
##  [112] 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2
##  [149] 2 2 2 2 2 1 2 2 2 2 1 2 2 2 2 2 2 1 2 1 1 2 1 2 2 2 2 2 1 1 2 1 1 1 2 2 2
##  [186] 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 1 1 1 2 2 2 2
##  [223] 2 2 1 2 2 2 2 1 1 2 2 2 2 2 1 1 2 2 1 2 2 2 2 2 2 1 2 2 2 1 2 2 2 2 2 2 2
##  [260] 2 2 1 1 2 2 2 2 2 2 2 1 1 1 2 2 2 2 1 2 2 2 2 1 2 2 1 2 2 2 2 2 1 2 2 2 2
##  [297] 1 2 1 2 1 2 2 2 2 2 2 2 1 2 2 2 2 1 1 1 2 2 2 2 2 2 2 2 2 2 1 2 2 1 2 2 2
##  [334] 2 1 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 1 2 2 2 2 2 1 2 2 2 2 1 2 2 2 1 1 1 2 2
##  [371] 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 1 2 2 2 2 2 1 2 2 2 2 2 2 1 2 2 1 2 2
##  [408] 2 1 1 1 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 1 2 2 2 2 1 2 2 2 2 1
##  [445] 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 1 1 1 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2
##  [482] 2 2 2 2 2 2 2 2 1 2 2 2 2 1 1 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 1 2 2 2 2
##  [519] 2 2 1 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2
##  [556] 2 2 2 2 2 2 2 2 1 2 2 2 2 2 1 2 2 2 1 1 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2
##  [593] 2 2 2 2 2 1 2 2 2 2 1 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 1 2 2
##  [630] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 1 2 2 2 2 1 2 2 2 2 2 2 2 2 2 1 2 2
##  [667] 2 2 2 2 2 2 2 2 1 1 2 2 1 2 2 2 2 2 2 2 1 1 1 2 2 2 1 1 2 2 2 2 1 1 2 2 2
##  [704] 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 1 2 2 2 2 2 2 1 2 2 2 2 2 1 2 2
##  [741] 2 2 1 2 1 2 2 1 1 2 2 2 2 1 1 2 2 2 2 1 2 2 1 1 1 2 2 2 2 2 1 1 2 2 2 1 2
##  [778] 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 1
##  [815] 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 1 2 1 2 2 2 2 1
##  [852] 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2
##  [889] 2 2 2 2 2 2 2 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 1 2 1 2 2 2 2 2 1
##  [926] 2 2 2 2 2 1 2 2 1 2 2 2 2 2 1 2 2 2 2 2 1 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 1
##  [963] 2 2 2 2 1 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 1 2 1 2 2 1 2
## [1000] 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 1 2 2 2 1 2 2 2 1 1 2 2
## [1037] 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2
## [1074] 2 2 2 2 2 1 1 2 2 2 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2
## [1111] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1
## [1148] 2 1 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1185] 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2
## [1222] 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 1 2 2 2 2
## [1259] 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1296] 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1333] 1 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 1 2 2 1 2 2 2 2 1 2 2 2 2 1 2
## [1370] 2 2 1 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 1 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2
## [1407] 2 2 2 2 2 2 1 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 2 2 2 2 2 1 2 2 2 2 2
## [1444] 2 1 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2
## [1481] 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1518] 2 1 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 1 2
## [1555] 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1
## [1592] 2 2 2 2 1 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2
## [1629] 2 2 1 1 2 2 2 1 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1 2 2 2 1 2 2 2 2 2 1
## [1666] 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1703] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2
## [1740] 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2
## [1777] 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1814] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1851] 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1888] 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 1 2
## [1925] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1962] 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1999] 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2036] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 1 2 2
## [2073] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2
## [2110] 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2147] 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2184] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2
## [2221] 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 1 2 2 2 2 1
## [2258] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2295] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2
## [2332] 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2369] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2
## [2406] 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2443] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1
## [2480] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2517] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2554] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2591] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2
## [2628] 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2665] 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2702] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2739] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2776] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2
## [2813] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2850] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2887] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2924] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2
## [2961] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2998] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [3035] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [3072] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [3109] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [3146] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [3183] 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [3220] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [3257] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [3294] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [3331] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [3368] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## 
## Within cluster sum of squares by cluster:
## [1] 4728.549 1421.516
##  (between_SS / total_SS =  39.7 %)
## 
## Available components:
## 
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
# groups of customers comparison
groups$size             # 303, 3100
```

```
## [1]  303 3100
```

```
groups$withinss         # 4728.549, 1421.516
```

```
## [1] 4728.549 1421.516
```

```
length(groups$cluster)  # 3403
```

```
## [1] 3403
```

```
# First group is much smaller than the second group. Sum of squares of elements
# in first group seems to be much greater that the one in the second group.


# create datasets of data from groups
clusteredData <- data.frame(clusteringData, groups$cluster)
str(clusteredData)
```

```
## 'data.frame':    3403 obs. of  5 variables:
##  $ Customer_ID    : int  1656 2211 949 68 1154 950 67 1155 117 168 ...
##  $ Office.Supplies: num  173 341 3417 25021 724 ...
##  $ Furniture      : num  0 0 3478 10692 531 ...
##  $ Technology     : num  0 0 14392 19034 8808 ...
##  $ groups.cluster : int  2 2 2 1 2 2 2 2 2 1 ...
```

```
cluster1Data <- clusteredData[clusteredData$groups.cluster == 1,]
str(cluster1Data)
```

```
## 'data.frame':    303 obs. of  5 variables:
##  $ Customer_ID    : int  68 168 2576 912 2627 882 2412 3281 1791 1605 ...
##  $ Office.Supplies: num  25021 963 13016 13278 7078 ...
##  $ Furniture      : num  10692 6646 22960 22400 22175 ...
##  $ Technology     : num  19034 28057 27199 2983 17960 ...
##  $ groups.cluster : int  1 1 1 1 1 1 1 1 1 1 ...
```

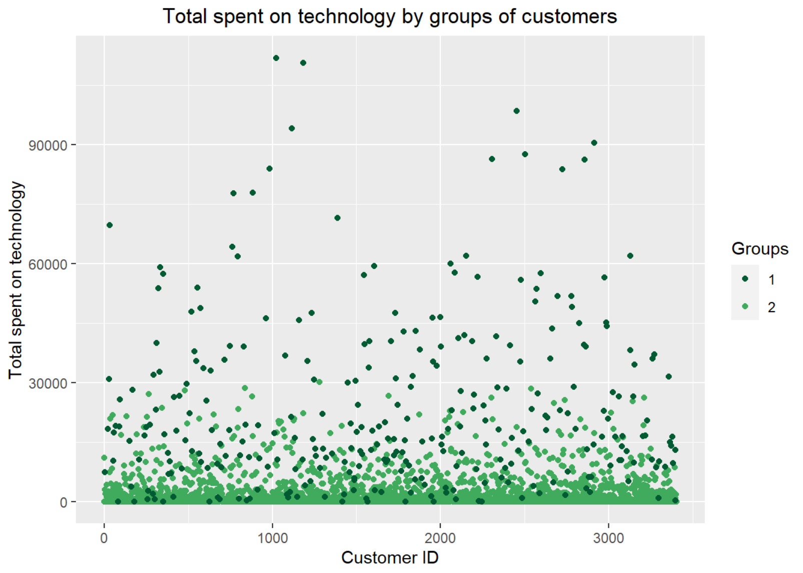
```
cluster2Data <- clusteredData[clusteredData$groups.cluster == 2,]
str(cluster2Data)
```

```
## 'data.frame':    3100 obs. of  5 variables:
##  $ Customer_ID    : int  1656 2211 949 1154 950 67 1155 117 1987 114 ...
##  $ Office.Supplies: num  173 341 3417 724 278 ...
##  $ Furniture      : num  0 0 3478 531 0 ...
##  $ Technology     : num  0 0 14392 8808 2224 ...
##  $ groups.cluster : int  2 2 2 2 2 2 2 2 2 2 ...
```

```
# plot clustered data
colnames(clusteredData)
```

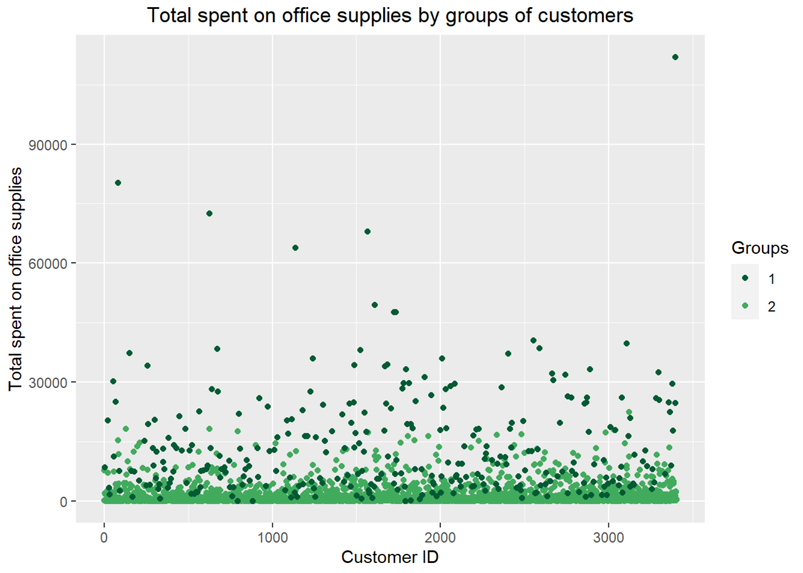
```
## [1] "Customer_ID"     "Office.Supplies" "Furniture"       "Technology"
## [5] "groups.cluster"
```

```
ggplot() +
  geom_point(data = cluster2Data,
             aes(x = Customer_ID, y = Technology, colour = "2")) +
  geom_point(data = cluster1Data,
            aes(x = Customer_ID, y = Technology, colour = "1")) +
  xlim(c(1, 3403)) +
  scale_colour_manual(values = rev(brewer.pal(name = "Greens", n = 8)[c(6, 8)])) +
  labs(x = "Customer ID", y = "Total spent on technology",
      title = "Total spent on technology by groups of customers",
      colour = "Groups") +
  theme(plot.title = element_text(hjust = 0.5))
```



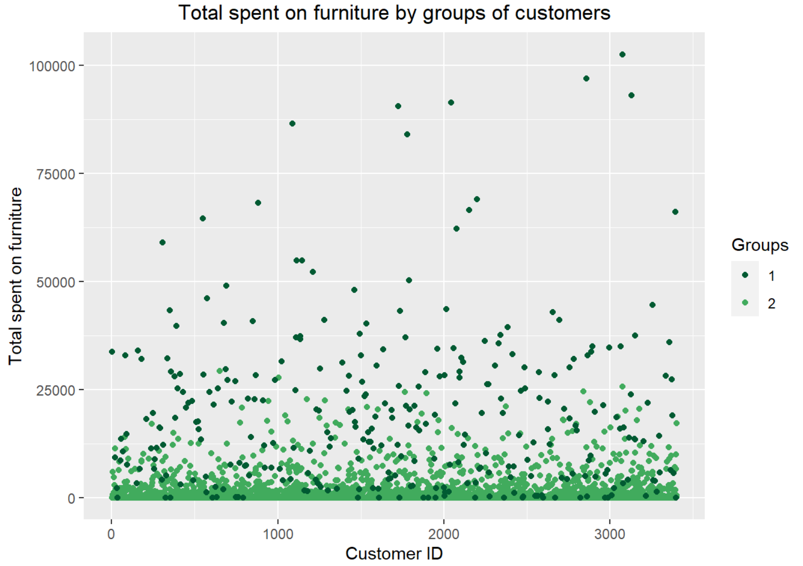
Total spent on technology by groups of customers

```
ggplot() +
  geom_point(data = cluster2Data,
             aes(x = Customer_ID, y = Office.Supplies, colour = "2")) +
  geom_point(data = cluster1Data,
             aes(x = Customer_ID, y = Office.Supplies, colour = "1")) +
  xlim(c(1, 3403)) +
  scale_colour_manual(values = rev(brewer.pal(name = "Greens", n = 8)[c(6, 8)])) +
  labs(x = "Customer ID", y = "Total spent on office supplies",
       title = "Total spent on office supplies by groups of customers",
       colour = "Groups") +
  theme(plot.title = element_text(hjust = 0.5))
```

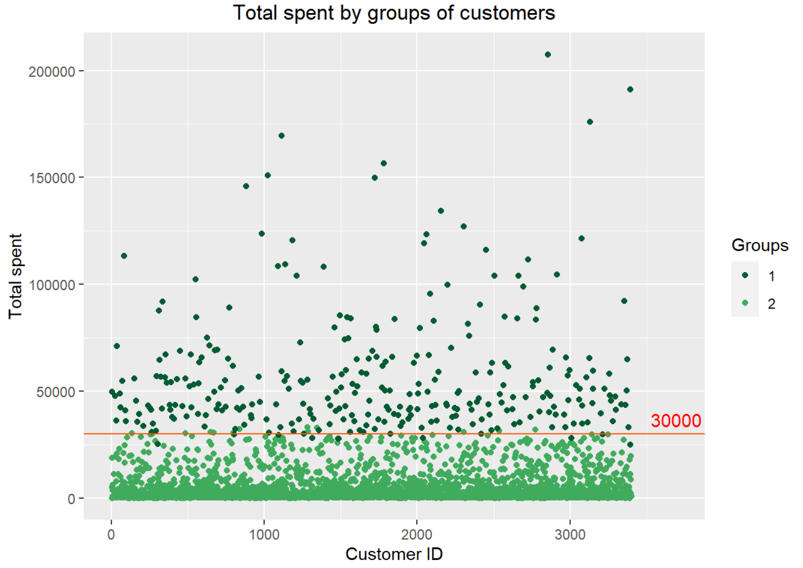


Total spent on office supplies by groups of customers

```
ggplot() +
  geom_point(data = cluster2Data,
             aes(x = Customer_ID, y = Furniture, colour = "2")) +
  geom_point(data = cluster1Data,
             aes(x = Customer_ID, y = Furniture, colour = "1")) +
  xlim(c(1, 3403)) +
  scale_colour_manual(values = rev(brewer.pal(name = "Greens", n = 8)[c(6, 8)])) +
  labs(x = "Customer ID", y = "Total spent on furniture",
       title = "Total spent on furniture by groups of customers",
       colour = "Groups") +
  theme(plot.title = element_text(hjust = 0.5))
```



```
ggplot() +
  geom_point(data = data.frame(ID = cluster2Data$Customer_ID,
                               TotalSpent = cluster2Data$Office.Supplies + cluster2Data$Furniture + cluster2Data$Technolog
y),
             aes(x = ID, y = TotalSpent, colour = "2")) +
  geom_point(data = data.frame(ID = cluster1Data$Customer_ID,
                               TotalSpent = cluster1Data$Office.Supplies + cluster1Data$Furniture + cluster1Data$Technolog
y),
             aes(x = ID, y = TotalSpent, colour = "1")) +
  geom_hline(aes(yintercept = 30000), col = "#FD5602") +
  geom_text(aes(x = c(3700), y = c(30000), label = "30000", vjust = -0.5), size = 4, col = "red") +
  xlim(c(1, 3700)) +
  scale_colour_manual(values = rev(brewer.pal(name = "Greens", n = 8)[c(6, 8)])) +
  labs(x = "Customer ID", y = "Total spent",
       title = "Total spent by groups of customers",
       colour = "Groups") +
  theme(plot.title = element_text(hjust = 0.5))
```



```
# As it can be seen from the plots, two groups of customers are are very
# different.
# First group consists of customers who spend large amounts of money
# buying products. They can be described as loyal customers and marketing
# products to them is not a priority. Customers from the first group usually
# spend more than 30000 on products.
# Second group consists of customers who spend smaller amounts of money on
# products. They are customers to whom we are not main suppliers and marketing
# products to them is a priority. Customers from the first group usually spend
# less than 30000 on products.



######################## LINEAR MODEL PROFIT PREDICTION ########################
# install.packages(c('corrplot', 'PerformanceAnalytics', 'vcd', 'MASS', 'leaps', 'caret', 'bootstrap'))
library('corrplot')
```

```
## Warning: package 'corrplot' was built under R version 4.1.3
```

```
## corrplot 0.92 loaded
```

```
library('PerformanceAnalytics')
```

```
## Warning: package 'PerformanceAnalytics' was built under R version 4.1.3
```

```
## Loading required package: xts
```

```
## Warning: package 'xts' was built under R version 4.1.3
```

```
## Loading required package: zoo
```

```
## Warning: package 'zoo' was built under R version 4.1.3
```

```
## 
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
## 
##     as.Date, as.Date.numeric
```

```
## 
## Attaching package: 'xts'
```

```
## The following objects are masked from 'package:dplyr':
## 
##     first, last
```

```
## 
## Attaching package: 'PerformanceAnalytics'
```

```
## The following object is masked from 'package:graphics':
## 
##     legend
```

```
library('vcd')
```

```
## Warning: package 'vcd' was built under R version 4.1.3
```

```
## Loading required package: grid
```

```
## 
## Attaching package: 'vcd'
```

```
## The following object is masked from 'package:PerformanceAnalytics':
## 
##     Kappa
```

```
library('MASS')
```

```
## Warning: package 'MASS' was built under R version 4.1.3
```

```
## 
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
## 
##     select
```

```
library('leaps')
```

```
## Warning: package 'leaps' was built under R version 4.1.3
```

```
library('caret')
```

```
## Warning: package 'caret' was built under R version 4.1.3
```

```
## Loading required package: lattice
```

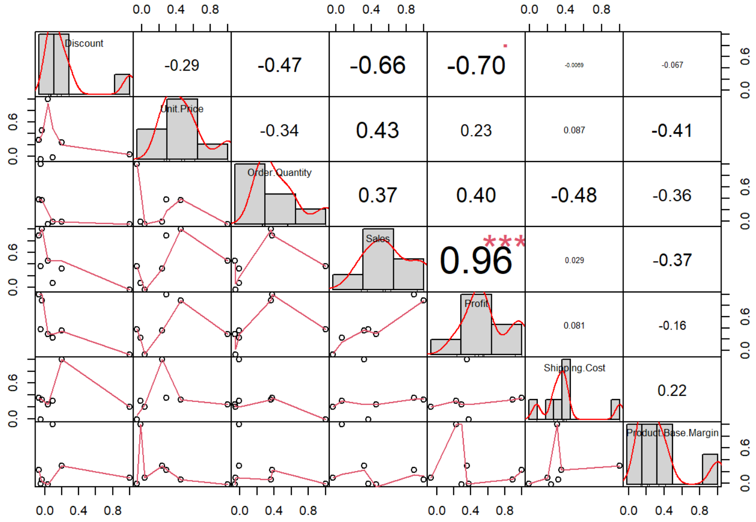
```
# Data selection and preparation
numericSalesData = subset(salesData[, sapply(salesData, is.numeric)], select = -Customer_ID)
factorSalesData = salesData[, sapply(salesData, is.factor)]
str(numericSalesData)
```

```
## 'data.frame':	16798 obs. of  7 variables:
##  $ Discount           : num  0.1 0.08 0.06 0 0.07 0.05 0.09 0.08 0.06 0.05 ...
##  $ Unit.Price         : num  6 96 41 292 101 155 9 15 41 155 ...
##  $ Order.Quantity     : int  32 2 3 4 43 32 16 43 1 8 ...
##  $ Sales              : num  173 177 116 1168 4039 ...
##  $ Profit             : num  106.4 45.6 33.9 605.1 2647.7 ...
##  $ Shipping.Cost      : num  5 35 3 49 45 7 2 2 3 7 ...
##  $ Product.Base.Margin: num  0.68 0.5 0.36 0.56 0.69 0.59 0.4 0.39 0.36 0.59 ...
```

```
cor(numericSalesData)
```

```
##                        Discount  Unit.Price Order.Quantity       Sales
## Discount             1.00000000  0.03566891   -0.047286423 -0.03736976
## Unit.Price           0.03566891  1.00000000   -0.055979845  0.45668317
## Order.Quantity      -0.04728642 -0.05597984    1.000000000  0.36716184
## Sales               -0.03736976  0.45668317    0.367161840  1.00000000
## Profit              -0.07047453  0.28811609    0.375928171  0.89280916
## Shipping.Cost        0.19879992  0.23927229   -0.009004473  0.32113750
## Product.Base.Margin  0.09145962 -0.01679710   -0.005333988  0.06831169
##                          Profit Shipping.Cost Product.Base.Margin
## Discount            -0.07047453   0.198799923         0.091459623
## Unit.Price           0.28811609   0.239272291        -0.016797103
## Order.Quantity       0.37592817  -0.009004473        -0.005333988
## Sales                0.89280916   0.321137502         0.068311693
## Profit               1.00000000   0.352106407         0.223957708
## Shipping.Cost        0.35210641   1.000000000         0.303872721
## Product.Base.Margin  0.22395771   0.303872721         1.000000000
```

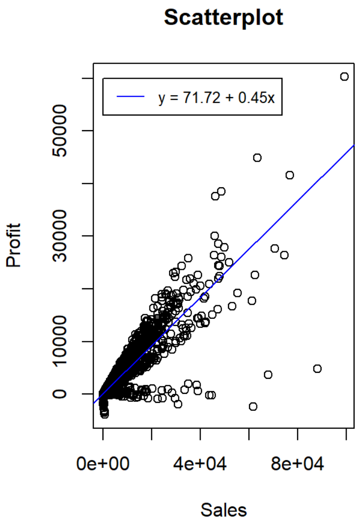
```
# Simple one variable linear model for Profit prediction
chart.Correlation(cor(numericSalesData))  # Profit ~ Sales -> 0.96***
```

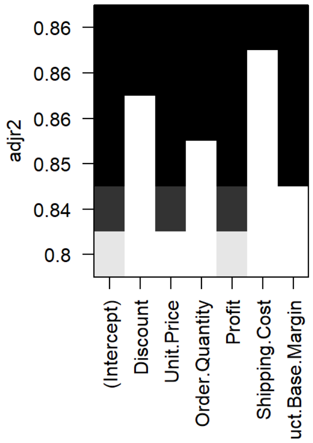


```
# correlation between Profit and Sales has the biggest correlation coefficient
# among all the variable combination, next closest is Profit Discount with
# -0.7 indicating a possible reversed correlation

# Train 80%, Test 20%
splitPercentage = 0.8
split <- sample(nrow(numericSalesData), splitPercentage * nrow(numericSalesData))
train <- numericSalesData[split, ]
test <- numericSalesData[-split, ]

fit <- lm(Profit ~ Sales, data=train)
summary(fit)  # Profit = 71.72 + 0.45 * Sales
```

```
## 
## Call:
## lm(formula = Profit ~ Sales, data = train)
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -35737    -87    -65      5  16277 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 62.347397   9.490837   6.569 5.24e-11 ***
## Sales        0.459303   0.001901 241.551  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 1028 on 13436 degrees of freedom
## Multiple R-squared:  0.8128, Adjusted R-squared:  0.8128 
## F-statistic: 5.835e+04 on 1 and 13436 DF,  p-value: < 2.2e-16
```

```
plot(train$Sales, train$Profit, xlab="Sales", ylab="Profit", main="Scatterplot")
abline(fit, col="Blue")
legend(0, 60000, legend=c("y = 71.72 + 0.45x"), col=c("blue"), lty=1:2, cex=0.8)

# Multiple variable regression, variable selection
leaps <- regsubsets(Sales ~ ., data=numericSalesData, nbest=1)
plot(leaps, scale="adjr2")
```





```
summary(leaps)
```

```
## Subset selection object
## Call: regsubsets.formula(Sales ~ ., data = numericSalesData, nbest = 1)
## 6 Variables  (and intercept)
##                      Forced in Forced out
## Discount                 FALSE      FALSE
## Unit.Price               FALSE      FALSE
## Order.Quantity           FALSE      FALSE
## Profit                   FALSE      FALSE
## Shipping.Cost            FALSE      FALSE
## Product.Base.Margin      FALSE      FALSE
## 1 subsets of each size up to 6
## Selection Algorithm: exhaustive
##          Discount Unit.Price Order.Quantity Profit Shipping.Cost
## 1  ( 1 ) " "      " "        " "            "*"    " "          
## 2  ( 1 ) " "      "*"        " "            "*"    " "          
## 3  ( 1 ) " "      "*"        " "            "*"    " "          
## 4  ( 1 ) " "      "*"        "*"            "*"    " "          
## 5  ( 1 ) "*"      "*"        "*"            "*"    " "          
## 6  ( 1 ) "*"      "*"        "*"            "*"    "*"          
##          Product.Base.Margin
## 1  ( 1 ) " "                
## 2  ( 1 ) " "                
## 3  ( 1 ) "*"                
## 4  ( 1 ) "*"                
## 5  ( 1 ) "*"                
## 6  ( 1 ) "*"
```

```
#          Discount Unit.Price Order.Quantity Profit Shipping.Cost Product.Base.Margin
# 1  ( 1 ) " "      " "        " "            "*"    " "           " "
# 2  ( 1 ) " "      "*"        " "            "*"    " "           " "
# 3  ( 1 ) " "      "*"        " "            "*"    " "           "*"
# 4  ( 1 ) " "      "*"        "*"            "*"    " "           "*"
# 5  ( 1 ) "*"      "*"        "*"            "*"    " "           "*"
# 6  ( 1 ) "*"      "*"        "*"            "*"    "*"           "*"

# Best single variable cor = Sales ~ Profit
# Best two variable cor = Sales ~ Profit + Unit.Price
# ...
# All variables included = Sales ~ .


# K-fold cross-validated R-square
shrinkage <- function(fit, k=10){
    require(bootstrap)

    # Fit and predict functions
    theta.fit <- function(x, y){lsfit(x, y)}
    theta.predict <- function(fit, x){cbind(1, x) %*% fit$coef}

    x <- fit$model[, 2:ncol(fit$model)]
    y <- fit$model[, 1]

    results <- crossval(x, y, theta.fit, theta.predict, ngroup=k)
    r2 <- cor(y, fit$fitted.values)**2  # Normal R2
    r2cv <- cor(y, results$cv.fit)**2   # Cross-validated R2

    cat("R-square =", r2, "\n")
    cat(k, "Fold Cross-Validated R-square =", r2cv, "\n")
    cat("Change =", r2 - r2cv, "\n")
}


# R-square = 0.7971082; acceptable R-square (~0.8)
# 10 Fold Cross-Validated R-square = 0.7945547
# Change = 0.002553464; small change
shrinkage(lm(Profit ~ Sales, data=numericSalesData))
```

```
## Loading required package: bootstrap
```

```
## R-square = 0.7971082
## 10 Fold Cross-Validated R-square = 0.7959051
## Change = 0.001203082
```

```
# R-square = 0.8151863
# 10 Fold Cross-Validated R-square = 0.8113877
# Change = 0.003798527
shrinkage(lm(Profit ~ Sales + Unit.Price, data=numericSalesData))
```

```
## R-square = 0.8151863
## 10 Fold Cross-Validated R-square = 0.8106929
## Change = 0.00449337
```

```
# R-square = 0.8396185
# 10 Fold Cross-Validated R-square = 0.8357494
# Change = 0.003869127
shrinkage(lm(Profit ~ Sales + Unit.Price + Product.Base.Margin, data=numericSalesData))
```

```
## R-square = 0.8396185
## 10 Fold Cross-Validated R-square = 0.8365046
## Change = 0.003113916
```

```
# R-square = 0.8401814
# 10 Fold Cross-Validated R-square = 0.836934
# Change = 0.003247305
shrinkage(lm(Profit ~ Sales + Unit.Price + Product.Base.Margin + Order.Quantity, data=numericSalesData))
```

```
## R-square = 0.8401814
## 10 Fold Cross-Validated R-square = 0.8360893
## Change = 0.004092098
```

```
# R-square = 0.8421473
# 10 Fold Cross-Validated R-square = 0.838756
# Change = 0.003391272
shrinkage(lm(Profit ~ Sales + Unit.Price + Product.Base.Margin + Order.Quantity + Discount, data=numericSalesData))
```

```
## R-square = 0.8421473
## 10 Fold Cross-Validated R-square = 0.8380442
## Change = 0.004103131
```

```
# R-square = 0.8447364; good R-square (~0.84)
# 10 Fold Cross-Validated R-square = 0.8416904
# Change = 0.003045998; small change
shrinkage(lm(Profit ~ ., data=numericSalesData))
```

```
## R-square = 0.8447364
## 10 Fold Cross-Validated R-square = 0.8417276
## Change = 0.003008779
```

```
# Factor correlation
summary(factorSalesData)
```

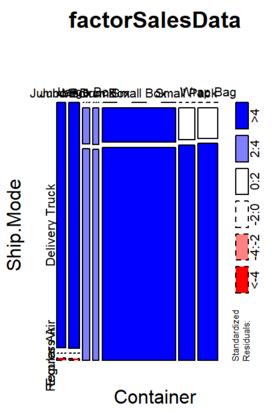
```
##        Order.Priority           Department          Container   
##  Critical     :3216   Furniture      :3448   Jumbo Box :1064  
##  High         :3536   Office Supplies:9220   Jumbo Drum:1248  
##  Low          :3440   Technology     :4130   Large Box : 812  
##  Medium       :3262                          Medium Box: 732  
##  Not Specified:3344                          Small Box :8694  
##                                              Small Pack:1912  
##                                              Wrap Bag  :2336  
##                                Category           Customer.Segment
##  Paper                           :2450   Consumer      :3298   
##  Binders and Binder Accessories:1830   Corporate     :6152   
##  Telephones and Communication  :1766   Home Office   :4064   
##  Office Furnishings            :1576   Small Business:3284   
##  Computer Peripherals          :1516                         
##  Pens & Art Supplies           :1266                         
##  (Other)                       :6394                         
##            Region                  State                      Country...Region
##  AsiaPac      :3802   California     : 1021   United States of America:9426  
##  EMEA         :1894   Texas          :  646   China                   :1257  
##  Latam        :1620   Illinois       :  584   India                   : 746  
##  North America:9482   New York       :  574   Brazil                  : 672  
##                       Florida        :  522   Japan                   : 507  
##                       Guangdong Sheng:  417   Mexico                  : 388  
##                       (Other)        :13034   (Other)                 :3802  
##            City                 Ship.Mode    
##  Guangzhou   :  357   Delivery Truck: 2292  
##  Buenos Aires:  341   Express Air   : 1966  
##  Seoul       :  292   Regular Air   :12540  
##  Tokyo       :  286                         
##  Paris       :  248                         
##  Beijing     :  245                         
##  (Other)     :15029                         
```

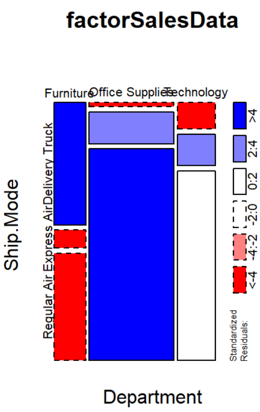
```
mosaicplot(Container ~ Ship.Mode, data=factorSalesData, shade=TRUE, legend=TRUE)
```

```
## Warning: In mosaicplot.default(table(mf), main = main, ...) :
##  extra argument 'legend' will be disregarded
```

```
mosaicplot(Department ~ Ship.Mode, data=factorSalesData, shade=TRUE, legend=TRUE)
```

```
## Warning: In mosaicplot.default(table(mf), main = main, ...) :
##  extra argument 'legend' will be disregarded
```





```
table(factorSalesData$Department, factorSalesData$Ship.Mode)
```

```
## 
##                   Delivery Truck Express Air Regular Air
##   Furniture                 1704         250        1494
##   Office Supplies            150        1188        7882
##   Technology                 438         528        3164
```

```
table(factorSalesData$Container, factorSalesData$Ship.Mode)
```

```
## 
##              Delivery Truck Express Air Regular Air
##   Jumbo Box            1054           0          10
##   Jumbo Drum           1238           0          10
##   Large Box               0         118         694
##   Medium Box              0         108         624
##   Small Box               0        1204        7490
##   Small Pack              0         248        1664
##   Wrap Bag                0         288        2048
```

```
# X-squared = 4912, df = 4, p-value < 2.2e-16
chisq.test(table(factorSalesData$Department, factorSalesData$Ship.Mode))
```

```
## 
##  Pearson's Chi-squared test
## 
## data:  table(factorSalesData$Department, factorSalesData$Ship.Mode)
## X-squared = 4912, df = 4, p-value < 2.2e-16
```

```
# Pearson's Chi-squared test for count data shows the test statistic (X-squared)
# is 4912, indicating a large difference between the expected and observed
# frequencies in the contingency table. The degrees of freedom are 4, which
# is calculated as the product of the number of levels minus one for each of the
# two variables in the contingency table. The p-value for the test is less than
# 2.2e-16, which is extremely small, suggesting strong evidence against the null
# hypothesis of independence between the two categorical variables.

# Therefore, we reject the null hypothesis and
# conclude that there is a significant association between the "Department" and
# "Ship.Mode" variables.


# X-squared = 16636, df = 12, p-value < 2.2e-16
chisq.test(table(factorSalesData$Container, factorSalesData$Ship.Mode))
```

```
## 
##  Pearson's Chi-squared test
## 
## data:  table(factorSalesData$Container, factorSalesData$Ship.Mode)
## X-squared = 16636, df = 12, p-value < 2.2e-16
```

```
# Similar to the previous test, this test also suggest correlation between the
# two variables. In this case "Container" and "Ship.Mode", with a larger
# degrees of freedom (12) and x-squared of 16636, a large difference between
# the expected and observed frequencies in the contingency table.
```