R Notebook

This is an R Markdown Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

```
setwd("C:/Users/willo/OneDrive/Documents/MSBA Program Classes/CRM Analytics")
options(repos = c(CRAN = "https://cran.rstudio.com"))

install.packages("tidyverse")
```

```
## Installing package into 'C:/Users/willo/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)

## package 'tidyverse' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\willo\AppData\Local\Temp\RtmpcT8nZb\downloaded_packages
```

```
install.packages("ggplot2")
```

```
## Installing package into 'C:/Users/willo/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)

## package 'ggplot2' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\willo\AppData\Local\Temp\RtmpcT8nZb\downloaded_packages
```

```
install.packages("car")
```

```
## Installing package into 'C:/Users/willo/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)

## package 'car' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\willo\AppData\Local\Temp\RtmpcT8nZb\downloaded_packages
```

```
install.packages("dplyr")
```

```
## Installing package into 'C:/Users/willo/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)
```

```
## package 'dplyr' successfully unpacked and MD5 sums checked

## Warning: cannot remove prior installation of package 'dplyr'

## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying
## C:\Users\willo\AppData\Local\R\win-library\4.3\00LOCK\dplyr\libs\x64\dplyr.dll
## to C:\Users\willo\AppData\Local\R\win-library\4.3\dplyr\libs\x64\dplyr.dll:
## Permission denied

## Warning: restored 'dplyr'

##
## The downloaded binary packages are in
##  C:\Users\willo\AppData\Local\Temp\RtmpcT8nZb\downloaded_packages
```

```
install.packages("beanplot")
```

```
## Installing package into 'C:/Users/willo/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)

## package 'beanplot' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\willo\AppData\Local\Temp\RtmpcT8nZb\downloaded_packages
```

```
install.packages("GGally")
```

```
## Installing package into 'C:/Users/willo/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)

## package 'GGally' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\willo\AppData\Local\Temp\RtmpcT8nZb\downloaded_packages
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(ggplot2)
library(car)
```

```
## Loading required package: carData
##
## Attaching package: 'car'
##
## The following object is masked from 'package:dplyr':
##
##     recode
##
## The following object is masked from 'package:purrr':
##
##     some
```

```r
library(dplyr)
library(beanplot)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```r
data <- read.csv("superstore dataset.csv")
#how many days is the product in the warehouse before shipping --> why is that
#happening?
install.packages("dplyr")
```

```
## Warning: package 'dplyr' is in use and will not be installed
```

```r
install.packages("lubridate")
```

```
## Warning: package 'lubridate' is in use and will not be installed
```

```r
library(dplyr)
library(lubridate)

head(data)
```

```
##   Row.ID        Order.ID Order.Date Ship.Date      Ship.Mode Customer.ID
## 1  42433    AG-2011-2040   1/1/2011  6/1/2011 Standard Class    TB-11280
## 2  22253   IN-2011-47883   1/1/2011  8/1/2011 Standard Class    JH-15985
## 3  48883    HU-2011-1220   1/1/2011  5/1/2011   Second Class      AT-735
## 4  11731 IT-2011-3647632   1/1/2011  5/1/2011   Second Class    EM-14140
## 5  22255   IN-2011-47883   1/1/2011  8/1/2011 Standard Class    JH-15985
## 6  22254   IN-2011-47883   1/1/2011  8/1/2011 Standard Class    JH-15985
##     Customer.Name     Segment        City           State   Country Postal.Code
## 1 Toby Braunhardt    Consumer Constantine     Constantine   Algeria          NA
## 2     Joseph Holt    Consumer Wagga Wagga New South Wales Australia          NA
## 3   Annie Thurman    Consumer    Budapest        Budapest   Hungary          NA
```

```
## 4    Eugene Moren Home Office   Stockholm       Stockholm    Sweden          NA
## 5     Joseph Holt    Consumer Wagga Wagga New South Wales Australia          NA
## 6     Joseph Holt    Consumer Wagga Wagga New South Wales Australia          NA
##   Market  Region       Product.ID        Category Sub.Category
## 1 Africa  Africa OFF-TEN-10000025 Office Supplies      Storage
## 2   APAC Oceania  OFF-SU-10000618 Office Supplies     Supplies
## 3   EMEA    EMEA OFF-TEN-10001585 Office Supplies      Storage
## 4     EU   North  OFF-PA-10001492 Office Supplies        Paper
## 5   APAC Oceania  FUR-FU-10003447       Furniture  Furnishings
## 6   APAC Oceania  OFF-PA-10001968 Office Supplies        Paper
##                            Product.Name   Sales Quantity Discount  Profit
## 1                   Tenex Lockers, Blue 408.300        2      0.0 106.140
## 2               Acme Trimmer, High Speed 120.366        3      0.1  36.036
## 3                Tenex Box, Single Width  66.120        4      0.0  29.640
## 4            Enermax Note Cards, Premium  44.865        3      0.5 -26.055
## 5             Eldon Light Bulb, Duo Pack 113.670        5      0.1  37.770
## 6 Eaton Computer Printout Paper, 8.5 x 11  55.242        2      0.1  15.342
##   Shipping.Cost Order.Priority
## 1         35.46         Medium
## 2          9.72         Medium
## 3          8.17           High
## 4          4.82           High
## 5          4.70         Medium
## 6          1.80         Medium
```

```
names(data)
```

```
##  [1] "Row.ID"         "Order.ID"       "Order.Date"     "Ship.Date"
##  [5] "Ship.Mode"      "Customer.ID"    "Customer.Name"  "Segment"
##  [9] "City"           "State"          "Country"        "Postal.Code"
## [13] "Market"         "Region"         "Product.ID"     "Category"
## [17] "Sub.Category"   "Product.Name"   "Sales"          "Quantity"
## [21] "Discount"       "Profit"         "Shipping.Cost"  "Order.Priority"
```

```
# Convert date columns to Date format
data$OrderDate <- as.Date(data$Order.Date, format = "%m/%d/%Y")
data$ShipDate <- as.Date(data$Ship.Date, format = "%m/%d/%Y")

# Calculating the days in the warehouse
data <- data %>%
  mutate(DaysInWarehouse = as.numeric(ShipDate - OrderDate))

# Summary of the days in the warehouse
summary(data$DaysInWarehouse)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##     0.0    61.0   122.0   108.9   153.0   214.0   37993
```

```
#Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
#0.0    61.0   122.0   108.9   153.0   214.0   37993
#About 25% of the orders were shipped within 61 days or less after the order
#date
```

```r
#Half of the orders took 122 days or fewer to ship, which means many orders
#experienced significant delays.
#On average, it took roughly 109 days for products to ship from the time they
#were ordered.
#75% of the orders were shipped within 153 days, but the remaining 25% took even
#longer than this.
#The longest recorded delay was 214 days, which is highly unusual and likely
#points to specific issues, such as inventory shortages, backorders, or
#operational inefficiencies.
#Investigate why delays are so frequent. Identify specific products, categories,
#or customers affected by long delays.
#Analyze the missing data. Determine whether missing dates are due to
#operational issues or incomplete record-keeping.
#Optimize logistics. Look into the shipping and inventory processes to reduce
#the time products spend in the warehouse.


#question 2- what is the most popular segment
# Count the occurrences of each segment
data %>%
  group_by(Segment) %>%
  summarise(TotalOrders = n()) %>%
  arrange(desc(TotalOrders))
```

```
## # A tibble: 3 x 2
##   Segment     TotalOrders
##   <chr>             <int>
## 1 Consumer          26518
## 2 Corporate         15429
## 3 Home Office        9343
```

```r
#question 3-Which countries have reoccurring customers
#count the number of orders per customer and filter for repeat customers
repeat_customers <- data %>%
  group_by(Country, Customer.ID) %>%
  summarise(OrderCount = n()) %>%
  filter(OrderCount > 1)
```

```
## `summarise()` has grouped output by 'Country'. You can override using the
## `.groups` argument.
```

```r
unique(repeat_customers$Country)
```

```
##   [1] "Afghanistan"                    "Albania"
##   [3] "Algeria"                        "Angola"
##   [5] "Argentina"                      "Armenia"
##   [7] "Australia"                      "Austria"
##   [9] "Azerbaijan"                     "Bahrain"
##  [11] "Bangladesh"                     "Barbados"
##  [13] "Belarus"                        "Belgium"
##  [15] "Benin"                          "Bolivia"
##  [17] "Bosnia and Herzegovina"         "Brazil"
```

```
##  [19] "Bulgaria"                    "Cambodia"
##  [21] "Cameroon"                    "Canada"
##  [23] "Central African Republic"    "Chad"
##  [25] "Chile"                       "China"
##  [27] "Colombia"                    "Cote d'Ivoire"
##  [29] "Croatia"                     "Cuba"
##  [31] "Czech Republic"              "Democratic Republic of the Congo"
##  [33] "Denmark"                     "Djibouti"
##  [35] "Dominican Republic"          "Ecuador"
##  [37] "Egypt"                       "El Salvador"
##  [39] "Equatorial Guinea"           "Eritrea"
##  [41] "Estonia"                     "Ethiopia"
##  [43] "Finland"                     "France"
##  [45] "Gabon"                       "Georgia"
##  [47] "Germany"                     "Ghana"
##  [49] "Guadeloupe"                  "Guatemala"
##  [51] "Guinea"                      "Guinea-Bissau"
##  [53] "Haiti"                       "Honduras"
##  [55] "Hong Kong"                   "Hungary"
##  [57] "India"                       "Indonesia"
##  [59] "Iran"                        "Iraq"
##  [61] "Ireland"                     "Israel"
##  [63] "Italy"                       "Jamaica"
##  [65] "Japan"                       "Jordan"
##  [67] "Kazakhstan"                  "Kenya"
##  [69] "Kyrgyzstan"                  "Lebanon"
##  [71] "Lesotho"                     "Liberia"
##  [73] "Libya"                       "Lithuania"
##  [75] "Macedonia"                   "Madagascar"
##  [77] "Malaysia"                    "Mali"
##  [79] "Martinique"                  "Mauritania"
##  [81] "Mexico"                      "Moldova"
##  [83] "Mongolia"                    "Montenegro"
##  [85] "Morocco"                     "Mozambique"
##  [87] "Myanmar (Burma)"             "Namibia"
##  [89] "Nepal"                       "Netherlands"
##  [91] "New Zealand"                 "Nicaragua"
##  [93] "Niger"                       "Nigeria"
##  [95] "Norway"                      "Pakistan"
##  [97] "Panama"                      "Papua New Guinea"
##  [99] "Paraguay"                    "Peru"
## [101] "Philippines"                 "Poland"
## [103] "Portugal"                    "Qatar"
## [105] "Republic of the Congo"       "Romania"
## [107] "Russia"                      "Rwanda"
## [109] "Saudi Arabia"                "Senegal"
## [111] "Sierra Leone"                "Singapore"
## [113] "Slovakia"                    "Slovenia"
## [115] "Somalia"                     "South Africa"
## [117] "South Korea"                 "Spain"
## [119] "Sri Lanka"                   "Sudan"
## [121] "Sweden"                      "Switzerland"
## [123] "Syria"                       "Taiwan"
## [125] "Tajikistan"                  "Tanzania"
```

```
## [127] "Thailand"                        "Togo"
## [129] "Trinidad and Tobago"             "Tunisia"
## [131] "Turkey"                          "Turkmenistan"
## [133] "Uganda"                          "Ukraine"
## [135] "United Arab Emirates"            "United Kingdom"
## [137] "United States"                   "Uruguay"
## [139] "Uzbekistan"                      "Venezuela"
## [141] "Vietnam"                         "Yemen"
## [143] "Zambia"                          "Zimbabwe"
```

```
#a list of 142 countries where customers have made multiple purchases.
#This indicates a broad international customer base with repeat business across
#diverse regions.

#question 4: What is the most popular category and subcategory?
# Group by Category and Sub-Category to find the most popular combinations
data %>%
  group_by(Category, Sub.Category) %>%
  summarise(TotalOrders = n()) %>%
  arrange(desc(TotalOrders))
```

```
## `summarise()` has grouped output by 'Category'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 17 x 3
## # Groups:   Category [3]
##    Category        Sub.Category TotalOrders
##    <chr>           <chr>              <int>
##  1 Office Supplies Binders             6152
##  2 Office Supplies Storage             5059
##  3 Office Supplies Art                 4883
##  4 Office Supplies Paper               3538
##  5 Furniture       Chairs              3434
##  6 Technology      Phones              3357
##  7 Furniture       Furnishings         3170
##  8 Technology      Accessories         3075
##  9 Office Supplies Labels              2606
## 10 Office Supplies Envelopes           2435
## 11 Office Supplies Supplies            2425
## 12 Office Supplies Fasteners           2420
## 13 Furniture       Bookcases           2411
## 14 Technology      Copiers             2223
## 15 Office Supplies Appliances          1755
## 16 Technology      Machines            1486
## 17 Furniture       Tables               861
```

```
#Office Supplies lead in orders, with high demand for Binders and Storage.
#Chairs dominate Furniture, while Phones and Accessories top Technology,
#signaling strong needs for essentials and communication tools. Lower orders for
#Tables and Machines suggest room for targeted promotions.

#question 5: Which country has the highest sales volume for certain
#subcategories?
```

```r
# Filter and group by Country and Sub-Category to get the sales volume
data %>%
  group_by(Country, Sub.Category) %>%
  summarise(SalesVolume = sum(Sales)) %>%
  arrange(desc(SalesVolume))
```

```
## `summarise()` has grouped output by 'Country'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 1,966 x 3
## # Groups:   Country [147]
##    Country       Sub.Category SalesVolume
##    <chr>         <chr>              <dbl>
##  1 United States Phones           330007.
##  2 United States Chairs           328449.
##  3 United States Storage          223844.
##  4 United States Tables           206966.
##  5 United States Binders          203413.
##  6 United States Machines         189239.
##  7 United States Accessories      167380.
##  8 United States Copiers          149528.
##  9 Australia     Chairs           142471.
## 10 Australia     Copiers          138261.
## # i 1,956 more rows
```

```r
#The USA leads in sales across multiple subcategories, especially in
#Phones, Chairs, and Storage, indicating broad demand. Australia
#shows strong sales in Chairs and Copiers, suggesting a focus on
#furniture and office equipment. This highlights opportunities to tailor
#strategies to each market's needs.

#question 6: What is the average profit per order size?
# Calculate profit per order and then find the average
data %>%
  group_by(Order.ID) %>%
  summarise(ProfitPerOrder = sum(Profit)) %>%
  summarise(AverageProfit = mean(ProfitPerOrder))
```

```
## # A tibble: 1 x 1
##   AverageProfit
##           <dbl>
## 1          58.6
```

```r
#Average Profit per Order: $58.60
#A relatively stable average profit indicates that most products and customers
#contribute similarly to profits, with fewer outliers or extreme variations in
#order profitability.

#question 7-How profitable are the top 10% of customers?
# Calculate total profit per customer
customer_profit <- data %>%
  group_by(Customer.ID) %>%
```

```
  summarise(TotalProfit = sum(Profit)) %>%
  arrange(desc(TotalProfit))

# Find the cutoff for the top 10% customers
cutoff <- quantile(customer_profit$TotalProfit, 0.9)

# Identify the top 10% customers
top_10_percent <- customer_profit %>%
  filter(TotalProfit >= cutoff)
top_10_percent
```

```
## # A tibble: 159 x 2
##    Customer.ID TotalProfit
##    <chr>             <dbl>
##  1 TC-20980          8787.
##  2 RB-19360          8524.
##  3 SC-20095          8106.
##  4 BE-11335          7791.
##  5 HL-15040          7658.
##  6 AB-10105          6913.
##  7 SP-20920          6650.
##  8 HM-14860          6545.
##  9 TA-21385          6275.
## 10 SE-20110          5864.
## # i 149 more rows
```

```
#The top 10% of customers consists of 159 customers with their total profit
#contributions ranging from $5,864 to $8,787. The top-performing customers
#(like TC-20980 and RB-19360) contribute significantly to the company's
#profitability.

#Question 8: Which customers or customer segments have the highest CLV?
# Calculate CLV (total profit) per customer
data %>%
  group_by(Customer.ID, Segment) %>%
  summarise(CLV = sum(Profit)) %>%
  arrange(desc(CLV))
```

```
## `summarise()` has grouped output by 'Customer.ID'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 1,590 x 3
## # Groups:   Customer.ID [1,590]
##    Customer.ID Segment       CLV
##    <chr>       <chr>       <dbl>
##  1 TC-20980    Corporate   8787.
##  2 RB-19360    Consumer    8524.
##  3 SC-20095    Consumer    8106.
##  4 BE-11335    Home Office 7791.
##  5 HL-15040    Consumer    7658.
##  6 AB-10105    Consumer    6913.
##  7 SP-20920    Consumer    6650.
```

```
##  8 HM-14860    Corporate   6545.
##  9 TA-21385    Home Office 6275.
## 10 SE-20110    Consumer    5864.
## # i 1,580 more rows
```

```
#High-CLV customers are spread across Consumer, Corporate, and Home Office
#segments, highlighting the need for targeted strategies in each. Corporate
#clients offer bulk-order potential, Home Office benefits from remote work
#trends, and Consumers thrive with personalized promotions and loyalty programs.

#Question 9: Which regions or cities have the highest CLV?
# Calculate CLV per city
data %>%
  group_by(City, Region) %>%
  summarise(CLV = sum(Profit)) %>%
  arrange(desc(CLV))
```

```
## `summarise()` has grouped output by 'City'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 3,753 x 3
## # Groups:   City [3,636]
##    City          Region     CLV
##    <chr>         <chr>    <dbl>
##  1 New York City East    62037.
##  2 Los Angeles   West    30441.
##  3 Seattle       West    29156.
##  4 Managua       Central 17854.
##  5 San Francisco West    17507.
##  6 Sydney        Oceania 16003.
##  7 London        North   15605.
##  8 San Salvador  Central 15037.
##  9 Mexico City   North   13342.
## 10 Vienna        Central 13319.
## # i 3,743 more rows
```

```
#New York, Los Angeles, and Seattle lead in CLV, highlighting the East and
#West region as key profit drivers. Emerging markets like Managua and
#Mexico City show potential for growth, suggesting opportunities for targeted
#strategies in both established and developing markets.

#Question 10: What product combinations are commonly purchased by high CLV
#customers?
# Identify high CLV customers (using previous top 10% calculation)
high_clv_customers <- top_10_percent$Customer.ID

# Filter transactions by high CLV customers
high_clv_data <- data %>%
  filter(Customer.ID %in% high_clv_customers)

# Find common product combinations
high_clv_data %>%
  group_by(Product.Name) %>%
```

```
  summarise(Count = n()) %>%
  arrange(desc(Count))
```

```
## # A tibble: 3,051 x 2
##    Product.Name                                    Count
##    <chr>                                           <int>
##  1 Staples                                            45
##  2 Eldon File Cart, Single Width                      20
##  3 Avery Index Tab, Clear                             15
##  4 Hon Executive Leather Armchair, Adjustable         15
##  5 Ibico Index Tab, Clear                             14
##  6 Sanford Pencil Sharpener, Water Color              14
##  7 Binney & Smith Pencil Sharpener, Water Color       13
##  8 Hewlett Copy Machine, Color                        13
##  9 BIC Pencil Sharpener, Blue                         12
## 10 Cardinal Index Tab, Economy                        12
## # i 3,041 more rows
```

```
#High-CLV customers favor office essentials like staples, index tabs,
#and pencil sharpeners, along with premium furniture and equipment like
#leather chairs and copy machines. This suggests opportunities to offer
#product bundles, loyalty rewards, and targeted promotions to boost repeat
#purchases and customer retention.

#survival analysis
a <- as.Date(data$Order.Date, format = "%m/%d/%Y") # Produces NA when format
#is not "%m/%d/%Y"
b <- as.Date(data$Order.Date, format = "%d-%m-%Y") # Produces NA when format
#is not "%d-%m-%Y"
a[is.na(a)] <- b[!is.na(b)] # Combine both while keeping their ranks
data$Order.Date <- a

summary(data$Order.Date)
```

```
##         Min.      1st Qu.       Median         Mean      3rd Qu.         Max.
## "2011-01-01" "2012-06-06" "2013-06-26" "2013-04-30" "2014-04-30" "2014-12-31"
```

```
install.packages("dplyr")
```

```
## Warning: package 'dplyr' is in use and will not be installed
```

```
library(dplyr)

data %>%  group_by(Customer.ID) %>%  summarise(last.date = max(Order.Date),
                                                early.date =
                                              min(Order.Date)) ->cust.date
data.frame(cust.date) -> cust.date
cust.date$time <- cust.date$last.date - cust.date$early.date
head(cust.date)
```

```
##   Customer.ID  last.date early.date      time
```

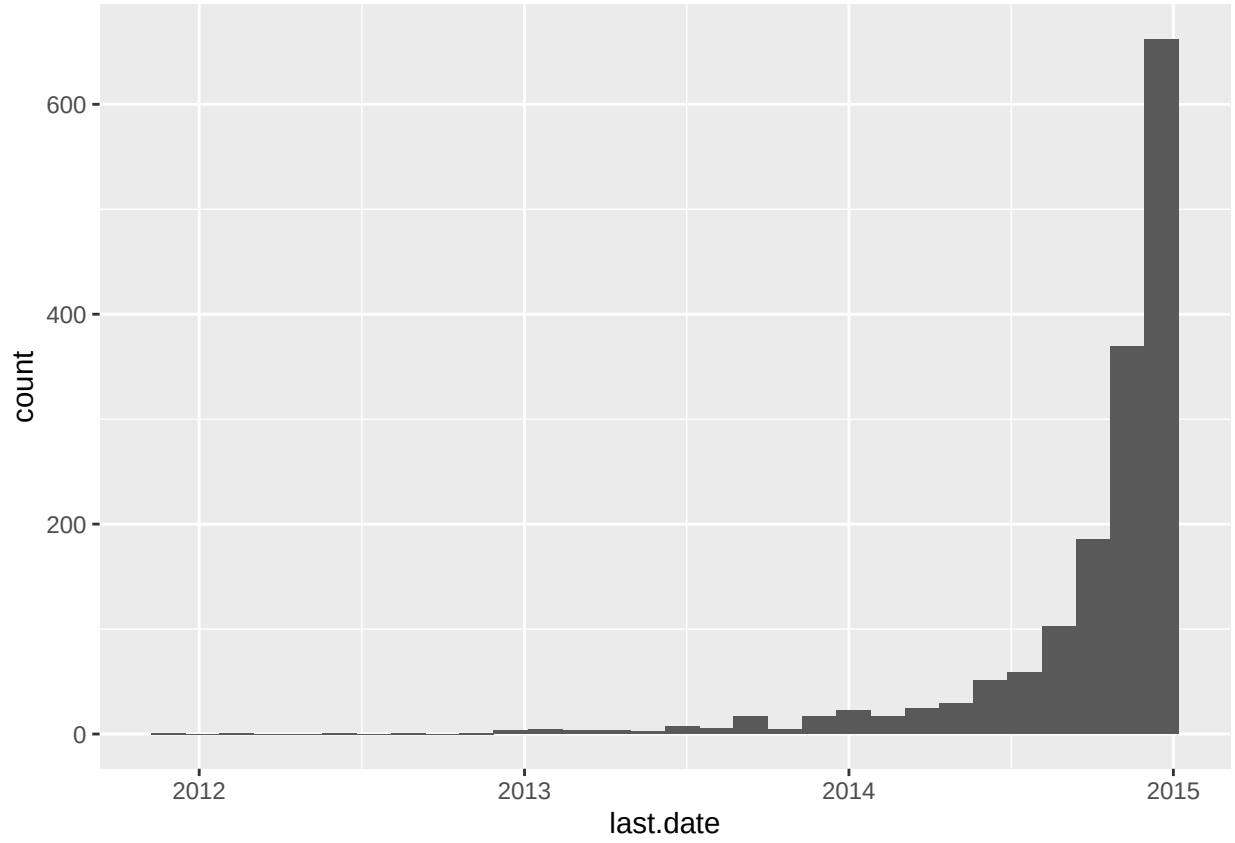
```
## 1    AA-10315 2014-12-23 2011-03-31 1363 days
## 2    AA-10375 2014-12-25 2011-04-21 1344 days
## 3    AA-10480 2014-09-05 2011-01-11 1333 days
## 4    AA-10645 2014-12-05 2011-01-12 1423 days
## 5      AA-315 2014-12-29 2011-08-06 1241 days
## 6      AA-375 2014-07-03 2011-01-06 1274 days
```

```
dim(cust.date)
```

```
## [1] 1590    4
```

```
ggplot(cust.date, aes(x = last.date)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
summary(cust.date$last.date)
```

```
##         Min.      1st Qu.       Median         Mean      3rd Qu.         Max.
## "2011-12-09" "2014-09-19" "2014-11-21" "2014-10-04" "2014-12-17" "2014-12-31"
```

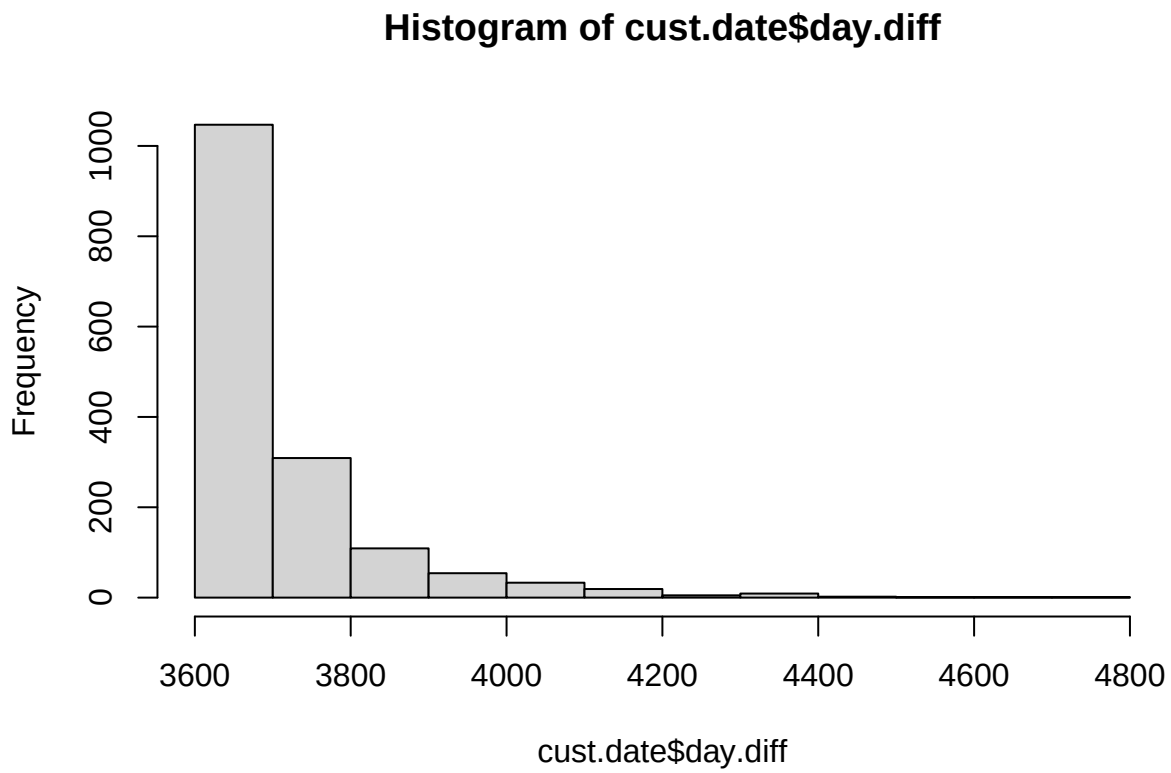
```
cust.date$day.diff <- as.numeric(difftime(Sys.Date(), cust.date$last.date,
                                          units =
                                            "days"))
summary(cust.date$day.diff)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    3626    3640    3666    3714    3729    4744
```

```
cust.date %>% mutate(state = ifelse(day.diff >= 3700, 1, 0)) -> cust.date
table(cust.date$state)
```

```
##
##    0    1
## 1047  543
```

```
hist(cust.date$day.diff)
```




```
today <- Sys.Date()
date_3700_days_ago <- today - 3700
date_3700_days_ago
```

```
## [1] "2014-10-18"
```

```
# Print the result
print(date_3700_days_ago)
```

```
## [1] "2014-10-18"
```

```r
#The histogram shows a large number of customers clustered around 3,600 days
#since their last order (about 10 years). This spike suggests that a significant
#portion of the customer base has been inactive for a long time, pointing to a
#potential issue with long-term retention.
#long tail of inactivity
install.packages('survival')
```

```
## Installing package into 'C:/Users/willo/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)
```

```
## package 'survival' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\willo\AppData\Local\Temp\RtmpcT8nZb\downloaded_packages
```

```r
install.packages('survminer')
```

```
## Installing package into 'C:/Users/willo/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)
```

```
## package 'survminer' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\willo\AppData\Local\Temp\RtmpcT8nZb\downloaded_packages
```

```r
library(survminer)
```

```
## Loading required package: ggpubr
```

```r
library(survival)
```

```
##
## Attaching package: 'survival'
##
## The following object is masked from 'package:survminer':
##
##     myeloma
```

```r
install.packages('geepack')
```

```
## Installing package into 'C:/Users/willo/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)
```

```
## package 'geepack' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\willo\AppData\Local\Temp\RtmpcT8nZb\downloaded_packages
```

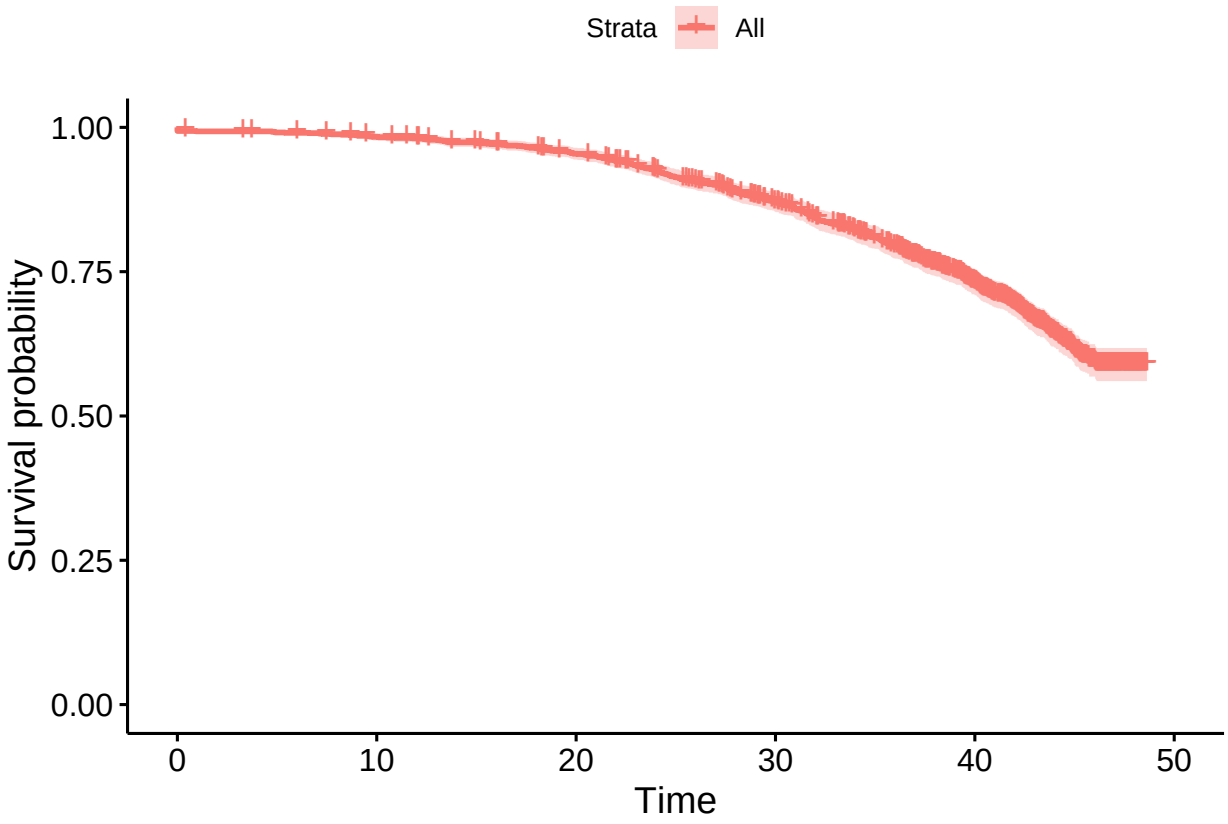
```r
library(geepack)
#Kaplan-Meier
library(survival)
cust.date$survival <- Surv(cust.date$time / 30, cust.date$state)

fit <- survfit(survival ~ 1, data = cust.date)
install.packages("survminer")
```

```
## Warning: package 'survminer' is in use and will not be installed
```

```r
library(survminer)


ggsurvplot(fit)
```




```r
#The survival curves indicate that customer retention rates differ significantly
#by market. Markets where the survival curve drops off sooner (showing a
#steep decline) have higher churn rates, meaning customers in these regions tend
#to churn earlier.

#Markets with flatter survival curves, such as EMEA or Canada, show better
#customer retention over time. These markets could potentially be leveraged as
#benchmarks for loyalty-building strategies or as lower priorities for retention
#interventions.
install.packages('survival')
```

```
## Warning: package 'survival' is in use and will not be installed
```

```
install.packages('survminer')
```

```
## Warning: package 'survminer' is in use and will not be installed
```

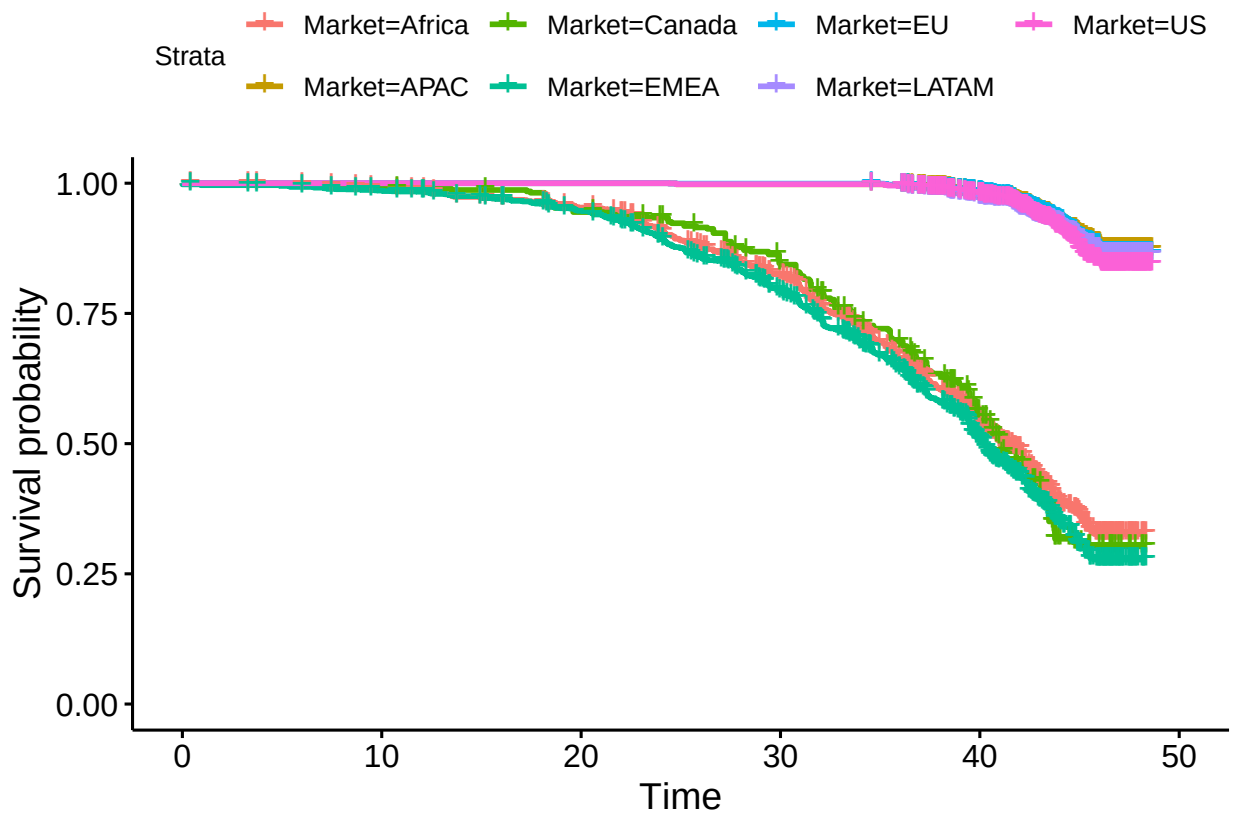
```
library(survminer)
library(dplyr)
library(survival)



cust.date %>% left_join(data) -> new.cust.date
```

```
## Joining with `by = join_by(Customer.ID)`
```

```
fit <- survfit(survival ~ Market, new.cust.date)
ggsurvplot(fit)
```




```
variable.names(new.cust.date)
```

```
##  [1] "Customer.ID"     "last.date"       "early.date"      "time"
```

```
##  [5] "day.diff"        "state"           "survival"        "Row.ID"
##  [9] "Order.ID"        "Order.Date"      "Ship.Date"       "Ship.Mode"
## [13] "Customer.Name"   "Segment"         "City"            "State"
## [17] "Country"         "Postal.Code"     "Market"          "Region"
## [21] "Product.ID"      "Category"        "Sub.Category"    "Product.Name"
## [25] "Sales"           "Quantity"        "Discount"        "Profit"
## [29] "Shipping.Cost"   "Order.Priority"  "OrderDate"       "ShipDate"
## [33] "DaysInWarehouse"
```

```
#The overall survival curve shows a steady decline in customer retention over
#time, meaning that as time progresses, a predictable portion of customers stop
#ordering.
#The narrower interval early on shows high confidence in initial retention rates
# while the wider interval later suggests increased variability as fewer
#customers remain active.

coxph(survival ~ Market + Sales + Segment, new.cust.date)
```

```
## Call:
## coxph(formula = survival ~ Market + Sales + Segment, data = new.cust.date)
##
##                           coef  exp(coef)   se(coef)       z        p
## MarketAPAC          -2.534e+00  7.936e-02  3.827e-02 -66.207  < 2e-16
## MarketCanada         3.797e-02  1.039e+00  7.319e-02   0.519   0.6040
## MarketEMEA           1.451e-01  1.156e+00  2.816e-02   5.154 2.55e-07
## MarketEU            -2.483e+00  8.350e-02  3.901e-02 -63.642  < 2e-16
## MarketLATAM         -2.431e+00  8.791e-02  3.779e-02 -64.344  < 2e-16
## MarketUS            -2.318e+00  9.851e-02  3.678e-02 -63.014  < 2e-16
## Sales                2.074e-05  1.000e+00  2.375e-05   0.873   0.3825
## SegmentCorporate    -2.275e-02  9.775e-01  2.405e-02  -0.946   0.3442
## SegmentHome Office   5.824e-02  1.060e+00  2.782e-02   2.093   0.0363
##
## Likelihood ratio test=12472  on 9 df, p=< 2.2e-16
## n= 51290, number of events= 9370
```

```
#Retention varies widely by market, with APAC, EU, LATAM, and US showing
#significantly lower churn rates (better retention) than other markets. In
#contrast, EMEA and Canada show higher churn risks, meaning customers in these
#regions might require more focused retention efforts.
#Sales do not significantly impact churn, suggesting that the amount a customer
#spends doesn't strongly predict their likelihood to stay or leave. Retention
#efforts might need to focus more on other factors, like regional strategies,
#rather than just sales volume.
#Corporate customers have slightly better retention,
#while Home Office customers have a higher likelihood of churn.
#This suggests that retention strategies might need to be more aggressive or
#tailored for Home Office customers.
#customer lifetime: The output shows that customers vary widely in tenure,
#with some staying for several years. This tenure information is critical in
#assessing retention: longer tenure implies stronger loyalty, while shorter
#tenure could indicate early churn.
```

```r
#CLV analysis
# Calculate AOV, purchase frequency, and customer lifetime
customer_metrics <- data %>%
  group_by(Customer.ID) %>%
  summarise(
    total_sales = sum(Sales),
    total_profit = sum(Profit),
    num_orders = n(),
    first_order_date = min(Order.Date),
    last_order_date = max(Order.Date)
  ) %>%
  mutate(
    AOV = total_sales / num_orders,  # Average Order Value
    purchase_frequency = num_orders / as.numeric(difftime(last_order_date,
                                                          first_order_date,
                                                          units = "weeks")),
    # Frequency per week
    customer_lifetime_weeks = as.numeric(difftime(last_order_date,
                                                  first_order_date,
                                                  units = "weeks"))
  )
#CLV= AOV Purchase * Frequency * Customer Lifetime
customer_metrics <- customer_metrics %>%
  mutate(
    CLV = AOV * purchase_frequency * customer_lifetime_weeks
  )
#result
head(customer_metrics)
```

```
## # A tibble: 6 x 10
##   Customer.ID total_sales total_profit num_orders first_order_date
##   <chr>             <dbl>        <dbl>      <int> <date>
## 1 AA-10315         13747.        448.          42 2011-03-31
## 2 AA-10375          5884.        677.          42 2011-04-21
## 3 AA-10480         17696.       1516.          38 2011-01-11
## 4 AA-10645         15344.       3051.          73 2011-01-12
## 5 AA-315            2243.        536.           8 2011-08-06
## 6 AA-375             654.         77.4         13 2011-01-06
## # i 5 more variables: last_order_date <date>, AOV <dbl>,
## #   purchase_frequency <dbl>, customer_lifetime_weeks <dbl>, CLV <dbl>
```

```r
#The output provides key metrics for the first six customers, showing their
#total sales, profit, number of orders, average order value (AOV), and purchase
#frequency over time. It also captures the duration of each customer's
#relationship with the company (in weeks) and their estimated Customer
#Lifetime Value (CLV). High AOV and CLV values indicate valuable customers,
#while purchase frequency reveals how often they engage.

#Calculate CLV Based on Profit
customer_metrics <- customer_metrics %>%
  mutate(
    Profit_CLV = (total_profit / num_orders) * purchase_frequency *
      customer_lifetime_weeks
```

```
  )

#result
head(customer_metrics)
```

```
## # A tibble: 6 x 11
##   Customer.ID total_sales total_profit num_orders first_order_date
##   <chr>             <dbl>        <dbl>      <int> <date>
## 1 AA-10315         13747.        448.          42 2011-03-31
## 2 AA-10375          5884.        677.          42 2011-04-21
## 3 AA-10480         17696.       1516.          38 2011-01-11
## 4 AA-10645         15344.       3051.          73 2011-01-12
## 5 AA-315            2243.        536.           8 2011-08-06
## 6 AA-375             654.         77.4         13 2011-01-06
## # i 6 more variables: last_order_date <date>, AOV <dbl>,
## #   purchase_frequency <dbl>, customer_lifetime_weeks <dbl>, CLV <dbl>,
## #   Profit_CLV <dbl>
```

```
#The output reveals that customer spending, order frequency, and lifetime value
#vary widely across the sample. High-value customers, such as `AA-10480` and
#`AA-10645`, have substantial total sales, profit, and CLV, indicating frequent
#and high-value purchases over an extended period. Conversely, customers like
#`AA-375` show lower AOV, profit, and CLV, suggesting infrequent and smaller
#orders. The profit-based CLV (Profit_CLV) highlights profitability differences,
#helping to identify which customers are most beneficial to retain for
#maximizing profit. Overall, this data suggests that focusing retention efforts
#on customers with higher CLV and Profit_CLV could drive significant long-term
#value for the company.

#linear model

library(tidyverse)
data$OrderDate <- as.Date(data$Order.Date, format = "%m/%d/%Y")
data$ShipDate <- as.Date(data$Ship.Date, format = "%m/%d/%Y")
data <- data %>% mutate(DaysInWarehouse = as.numeric(ShipDate - OrderDate))

#predict Profit based on DaysInWarehouse, Segment, and Category
model <- lm(Profit ~ DaysInWarehouse + Segment + Category, data = data)

summary(model)
```

```
##
## Call:
## lm(formula = Profit ~ DaysInWarehouse + Segment + Category, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6659.5   -27.3   -11.0    10.5  8328.6
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              25.852344   2.996556   8.627  < 2e-16 ***
## DaysInWarehouse           0.012944   0.008955   1.446   0.1483
```

```
## SegmentCorporate           5.722533   2.741566   2.087   0.0369 *
## SegmentHome Office         7.239170   3.283988   2.204   0.0275 *
## CategoryOffice Supplies -12.442940   3.144728  -3.957 7.62e-05 ***
## CategoryTechnology       36.992322   3.848001   9.613  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 168.7 on 19828 degrees of freedom
##   (31456 observations deleted due to missingness)
## Multiple R-squared:  0.01313,    Adjusted R-squared:  0.01288
## F-statistic: 52.77 on 5 and 19828 DF,  p-value: < 2.2e-16
```

```
#This linear model shows that both customer segment and product category
#significantly impact profit. "Corporate" and "Home Office" segments add $5.72
#and $7.24 to profit.. Product categories also matter: "Technology"
#products increase profit by $36.99 on average, while "Office Supplies" reduce
#it by $12.44. Although `DaysInWarehouse` has a small positive effect, it's not
#statistically significant, suggesting delays don't heavily impact profit.
#Overall, the model explains only a small portion of profit variation,
#indicating other factors likely play a larger role.

#MODEL INSIGHTS:
#Both the "Corporate" and "Home Office" segments positively impact profit, with
#average increases of $5.72 and $7.24, respectively. This suggests that
#targeting these segments may yield higher profits compared to others.

#Although DaysInWarehouse has a small positive coefficient, it's not
#statistically significant, meaning delays do not strongly influence profit.
#This suggests that other operational factors may be more critical for
#profitability.

#With an Adjusted R-squared of 0.01288, the model explains only a small portion
#of the profit variability, implying that there are other unaccounted factors
#that significantly affect profit. Further exploration could focus on additional
#variables to enhance predictive accuracy.

#logit regression
library(dplyr)
library(tidyverse)
#define the binary outcome variable (1 if Profit is above the median, 0
#otherwise)
median_profit <- median(data$Profit, na.rm = TRUE)
data <- data %>% mutate(HighProfit = ifelse(Profit > median_profit, 1, 0))
#logistic regression model to predict HighProfit
logit_model <- glm(HighProfit ~ DaysInWarehouse + Segment + Category,
                   data = data,
                   family = binomial)

#model summary
summary(logit_model)
```

```
##
## Call:
## glm(formula = HighProfit ~ DaysInWarehouse + Segment + Category,
```

```
##     family = binomial, data = data)
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)               0.2473051  0.0358697   6.895  5.4e-12 ***
## DaysInWarehouse          -0.0002766  0.0001082  -2.557   0.0106 *
## SegmentCorporate          0.0206995  0.0331140   0.625   0.5319
## SegmentHome Office        0.0202858  0.0396644   0.511   0.6090
## CategoryOffice Supplies -0.5110684  0.0375868 -13.597  < 2e-16 ***
## CategoryTechnology        0.4586690  0.0472356   9.710  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 27494  on 19833  degrees of freedom
## Residual deviance: 26763  on 19828  degrees of freedom
##   (31456 observations deleted due to missingness)
## AIC: 26775
##
## Number of Fisher Scoring iterations: 4
```

```
#The logistic regression results highlight a few interesting points. First off,
#the time a product spends in the warehouse (DaysInWarehouse) slightly reduces
#the odds of it being high profit, but the effect is small, so while it's worth
#noting, it's not a huge game-changer. Segment (whether an order is for
#Corporate, Home Office, or Consumer) doesn't seem to impact profit odds much,
#which actually might mean we don't need to obsess over segment-specific
#targeting for profit improvement. However, product category is where things get
#interesting. Orders for Office Supplies are less likely to be high-profit,
#while Technology orders are solidly on the profitable side, way more so than
#Furniture, which serves as our baseline here. So, if there's any takeaway for
#strategy, it's that category focus - especially on high-margin products in
#Technology - might be more impactful than adjusting operations around warehouse
#time or customer segment.

#The logistic regression model tried to find factors that predict high
#customer lifetime value (CLV) using `total_sales`, `AOV` (average order value),
#`purchase_frequency`, and `customer_lifetime_weeks`. But none of these
#variables turned out to be statistically significant in predicting high-CLV
#status. The model's residual deviance was also unusually low, which might mean
#it's overfitting or having trouble converging. This could be because the chosen
#predictors aren't quite capturing what makes a customer high-CLV, possibly due
#to high multicollinearity or an imbalance in the high-CLV data itself. To get
#a better fit, we might want to add some new predictors, check for correlation
#among variables, and consider resampling to balance the high-CLV group.

#Recommendations for Model Improvement:
#Add or Test Additional Variables: Consider adding variables that could impact
#profitability, such as Order.Quantity, Sales, Discount, or geographic variables.
#Address Missing Data: The model has dropped a large number of observations due
#to missing variables (31,456 observations deleted). If possible, investigate
#and
#handle missing data to improve the model's representativeness.
```

```r
#Consider Interaction Terms: There may be interactions between Category and
#Segment or DaysInWarehouse and Category that could provide deeper insights.
#Conclusion: This output provides a reasonable starting point with meaningful
#findings related to DaysInWarehouse and product categories. However, it could
#benefit from refinement, particularly in exploring other predictive factors,
#handling missing data, and potentially adding interaction effects to capture
#more nuanced relationships.

#Logistic Regression for Predicting High-CLV Customers
median_clv <- median(customer_metrics$CLV, na.rm = TRUE)
customer_metrics <- customer_metrics %>%
  mutate(HighCLV = ifelse(CLV > median_clv, 1, 0))
logit_clv_model <- glm(HighCLV ~ total_sales + AOV + purchase_frequency +
                         customer_lifetime_weeks,
                       data = customer_metrics,
                       family = binomial)
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
summary(logit_clv_model)
```

```
##
## Call:
## glm(formula = HighCLV ~ total_sales + AOV + purchase_frequency +
##     customer_lifetime_weeks, family = binomial, data = customer_metrics)
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)             -2075.7941 28208.9290  -0.074    0.941
## total_sales                0.3793     5.3223   0.071    0.943
## AOV                       -0.3114     9.9245  -0.031    0.975
## purchase_frequency      -954.5700 32239.2656  -0.030    0.976
## customer_lifetime_weeks   -0.5983    17.6480  -0.034    0.973
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2.1903e+03  on 1579  degrees of freedom
## Residual deviance: 2.4170e-05  on 1575  degrees of freedom
##   (10 observations deleted due to missingness)
## AIC: 10
##
## Number of Fisher Scoring iterations: 25
```

```r
#dont like this model as much!

#linear regression model
library(dplyr)

linear_clv_model <- lm(CLV ~ total_sales + AOV + purchase_frequency +
                         customer_lifetime_weeks,
```

```
                      data = customer_metrics)

summary(linear_clv_model)
```

```
##
## Call:
## lm(formula = CLV ~ total_sales + AOV + purchase_frequency + customer_lifetime_weeks,
##     data = customer_metrics)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -4.613e-10 -1.400e-13  3.400e-13  7.200e-13  1.104e-11
##
## Coefficients:
##                            Estimate Std. Error    t value Pr(>|t|)
## (Intercept)              -1.318e-11  1.696e-12 -7.771e+00 1.39e-14 ***
## total_sales               1.000e+00  8.803e-17  1.136e+16  < 2e-16 ***
## AOV                       8.488e-16  4.071e-15  2.080e-01    0.835
## purchase_frequency       -2.085e-12  2.838e-12 -7.350e-01    0.463
## customer_lifetime_weeks -9.571e-15  9.367e-15 -1.022e+00    0.307
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.169e-11 on 1575 degrees of freedom
##   (10 observations deleted due to missingness)
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 1.389e+32 on 4 and 1575 DF,  p-value: < 2.2e-16
```

```
#The model suggests that total_sales is the primary driver of CLV, almost
#perfectly predicting it. However, the R-squared value of 1 and the small
#residuals indicate that the model may be overfitted or overly reliant on
#total_sales alone. The other variables (AOV, purchase_frequency, and
#customer_lifetime_weeks) don't have a significant impact, which might suggest
#multicollinearity or redundancy in the predictors. To improve this model, it
#might be helpful to examine correlations among the variables and consider
#removing or transforming predictors to ensure they offer unique insights into
#CLV.


#big basket analysis

#reference for doing group basket analyses
library(arules)
```

```
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
##
##
```

```
## Attaching package: 'arules'
##
## The following object is masked from 'package:car':
##
##     recode
##
## The following object is masked from 'package:dplyr':
##
##     recode
##
## The following objects are masked from 'package:base':
##
##     abbreviate, write
```

```
library(arulesViz)
#convert dataframe to data.matrix
arm <- read.csv("superstore dataset.csv")
dim(arm)
```

```
## [1] 51290    24
```

```
#51290 rows, 24 Cols,

dim(arm)
```

```
## [1] 51290    24
```

```
#51290 rows, 23 Cols, Order.ID is the key column.
str(arm)
```

```
## 'data.frame':    51290 obs. of  24 variables:
##  $ Row.ID        : int  42433 22253 48883 11731 22255 22254 21613 34662 44508 23688 ...
##  $ Order.ID      : chr  "AG-2011-2040" "IN-2011-47883" "HU-2011-1220" "IT-2011-3647632" ...
##  $ Order.Date    : chr  "1/1/2011" "1/1/2011" "1/1/2011" "1/1/2011" ...
##  $ Ship.Date     : chr  "6/1/2011" "8/1/2011" "5/1/2011" "5/1/2011" ...
##  $ Ship.Mode     : chr  "Standard Class" "Standard Class" "Second Class" "Second Class" ...
##  $ Customer.ID   : chr  "TB-11280" "JH-15985" "AT-735" "EM-14140" ...
##  $ Customer.Name : chr  "Toby Braunhardt" "Joseph Holt" "Annie Thurman" "Eugene Moren" ...
##  $ Segment       : chr  "Consumer" "Consumer" "Consumer" "Home Office" ...
##  $ City          : chr  "Constantine" "Wagga Wagga" "Budapest" "Stockholm" ...
##  $ State         : chr  "Constantine" "New South Wales" "Budapest" "Stockholm" ...
##  $ Country       : chr  "Algeria" "Australia" "Hungary" "Sweden" ...
##  $ Postal.Code   : int  NA NA NA NA NA NA NA 92691 NA NA ...
##  $ Market        : chr  "Africa" "APAC" "EMEA" "EU" ...
##  $ Region        : chr  "Africa" "Oceania" "EMEA" "North" ...
##  $ Product.ID    : chr  "OFF-TEN-10000025" "OFF-SU-10000618" "OFF-TEN-10001585" "OFF-PA-10001492" ...
##  $ Category      : chr  "Office Supplies" "Office Supplies" "Office Supplies" "Office Supplies" ...
##  $ Sub.Category  : chr  "Storage" "Supplies" "Storage" "Paper" ...
##  $ Product.Name  : chr  "Tenex Lockers, Blue" "Acme Trimmer, High Speed" "Tenex Box, Single Width" "T
##  $ Sales         : num  408.3 120.4 66.1 44.9 113.7 ...
##  $ Quantity      : int  2 3 4 3 5 2 2 2 1 3 ...
##  $ Discount      : num  0 0.1 0 0.5 0.1 0.1 0 0.15 0 0 ...
```

```
##  $ Profit        : num  106.1 36 29.6 -26.1 37.8 ...
##  $ Shipping.Cost : num  35.46 9.72 8.17 4.82 4.7 ...
##  $ Order.Priority: chr  "Medium" "Medium" "High" "High" ...
```

```
#Most of the cols are character/ Numeric data type.
#Our consideration will be subcategory
summary(arm)
```

```
##      Row.ID        Order.ID          Order.Date         Ship.Date
##  Min.   :    1   Length:51290       Length:51290       Length:51290
##  1st Qu.:12823   Class :character   Class :character   Class :character
##  Median :25646   Mode  :character   Mode  :character   Mode  :character
##  Mean   :25646
##  3rd Qu.:38468
##  Max.   :51290
##
##   Ship.Mode         Customer.ID        Customer.Name        Segment
##  Length:51290       Length:51290       Length:51290       Length:51290
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##      City              State              Country           Postal.Code
##  Length:51290       Length:51290       Length:51290       Min.   : 1040
##  Class :character   Class :character   Class :character   1st Qu.:23223
##  Mode  :character   Mode  :character   Mode  :character   Median :56431
##                                                           Mean   :55190
##                                                           3rd Qu.:90008
##                                                           Max.   :99301
##                                                           NA's   :41296
##     Market             Region           Product.ID          Category
##  Length:51290       Length:51290       Length:51290       Length:51290
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  Sub.Category       Product.Name           Sales              Quantity
##  Length:51290       Length:51290       Min.   :    0.444   Min.   : 1.000
##  Class :character   Class :character   1st Qu.:   30.759   1st Qu.: 2.000
##  Mode  :character   Mode  :character   Median :   85.053   Median : 3.000
##                                        Mean   :  246.491   Mean   : 3.477
##                                        3rd Qu.:  251.053   3rd Qu.: 5.000
##                                        Max.   :22638.480   Max.   :14.000
##
##     Discount          Profit         Shipping.Cost    Order.Priority
##  Min.   :0.0000   Min.   :-6599.98   Min.   :  0.00   Length:51290
##  1st Qu.:0.0000   1st Qu.:    0.00   1st Qu.:  2.61   Class :character
##  Median :0.0000   Median :    9.24   Median :  7.79   Mode  :character
##  Mean   :0.1429   Mean   :   28.61   Mean   : 26.38
##  3rd Qu.:0.2000   3rd Qu.:   36.81   3rd Qu.: 24.45
```

```
##  Max.   :0.8500   Max.   : 8399.98   Max.   :933.57
##
```

```r
nrow(unique(arm))
```

```
## [1] 51290
```

```r
#When grouped together, all the records are unique. Number of records - 51290
#Lets find unique order id count
length(unique(arm$Order.ID))
```

```
## [1] 25035
```

```r
#25035
length(unique(arm$Sub.Category))
```

```
## [1] 17
```

```r
#17 unique sub category
#25035 unique Order.ID.
#The dataset contains 51,290 unique rows and 23 columns after removing the
#unnecessary Row.ID column. Each row represents a transaction, with Order.ID
#grouping multiple rows for the same order (25,035 unique orders).
#Most variables are character (e.g., Sub.Category, Market) or numeric
#(e.g., Sales, Profit). The Postal.Code column has substantial missing data
#(41,296 values).
#Key metrics include Sales, ranging from 0.444 to 22,638.48 (median: 85.05),
#and Profit, with significant variability from -6,599.98 to 8,399.98
#(median: 9.24). Discounts are generally low, averaging 0.1429. The dataset
#features 17 unique Sub.Category values, capturing diverse product groups like
#"Storage" and "Supplies."
#This data provides a robust foundation for analyzing sales, profitability, and
#customer behavior across products and regions. It highlights trends in profit
#margins, the impact of discounts, and opportunities for market segmentation.

install.packages("arules")
```

```
## Warning: package 'arules' is in use and will not be installed
```

```r
library(arules)


arm_mini <- arm[,c("Order.ID","Sub.Category")]
head(arm_mini)
```

```
##          Order.ID Sub.Category
## 1    AG-2011-2040      Storage
## 2   IN-2011-47883     Supplies
## 3    HU-2011-1220      Storage
## 4 IT-2011-3647632        Paper
## 5   IN-2011-47883  Furnishings
## 6   IN-2011-47883        Paper
```

```r
write.csv(arm_mini, "transdata", row.names = F) #force the dataframe into csv
transdata <- read.transactions(
  file = "transdata",
  format = "single",
  sep = ",",
  cols = c("Order.ID", "Sub.Category"),
  rm.duplicates = TRUE,
  header = TRUE
)
class(transdata)
```

```
## [1] "transactions"
## attr(,"package")
## [1] "arules"
```

```r
arm_transactions <- transdata
summary(arm_transactions)
```

```
## transactions as itemMatrix in sparse format with
##  25035 rows (elements/itemsets/transactions) and
##  17 columns (items) and a density of 0.1113805
##
## most frequent items:
## Binders Storage     Art   Paper  Chairs (Other)
##    5392    4534    4366    3234    3187   26690
##
## element (itemset/transaction) length distribution:
## sizes
##     1     2     3     4     5     6     7     8     9    10    11
## 12800  6469  3193  1484   626   304   101    42     9     5     2
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   1.000   1.000   1.893   2.000  11.000
##
## includes extended item information - examples:
##        labels
## 1 Accessories
## 2  Appliances
## 3         Art
##
## includes extended transaction information - examples:
##   transactionID
## 1  AE-2011-9160
## 2  AE-2013-1130
## 3  AE-2013-1530
```

```r
inspect(arm_transactions[1:5])
```

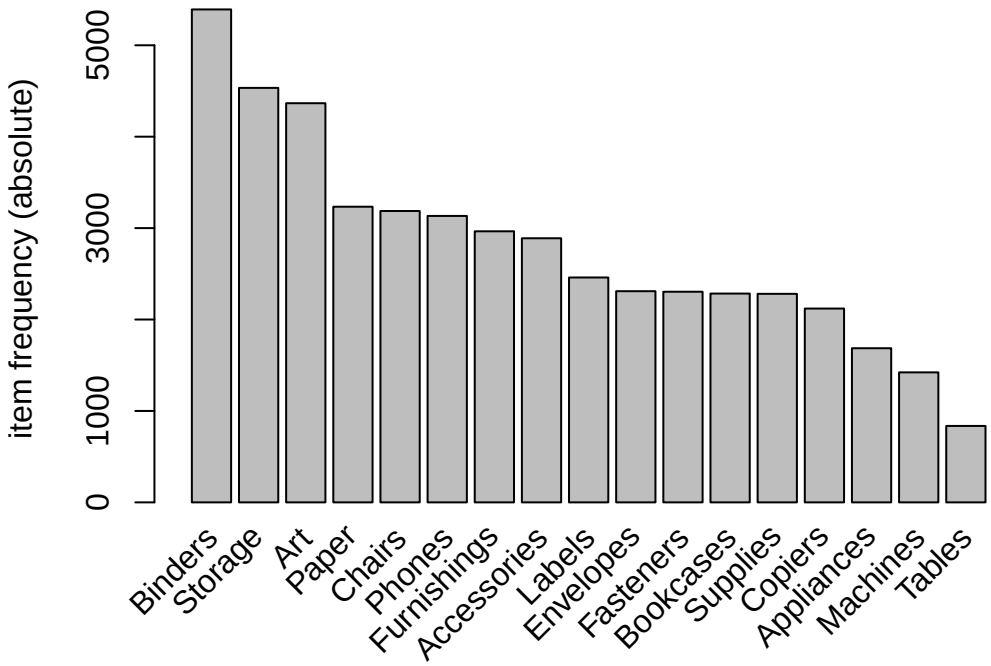
```
##     items                          transactionID
## [1] {Machines, Storage}            AE-2011-9160
## [2] {Bookcases, Fasteners}         AE-2013-1130
## [3] {Storage, Supplies}            AE-2013-1530
```

```
## [4] {Storage}                         AE-2014-2840
## [5] {Art, Binders, Phones, Storage} AE-2014-3830
```

```
arm_transactions<-as(transdata,"transactions")
summary(arm_transactions)
```

```
## transactions as itemMatrix in sparse format with
##  25035 rows (elements/itemsets/transactions) and
##  17 columns (items) and a density of 0.1113805
##
## most frequent items:
## Binders Storage     Art   Paper  Chairs (Other)
##    5392    4534    4366    3234    3187   26690
##
## element (itemset/transaction) length distribution:
## sizes
##     1     2     3     4     5     6     7     8     9    10    11
## 12800  6469  3193  1484   626   304   101    42     9     5     2
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   1.000   1.000   1.893   2.000  11.000
##
## includes extended item information - examples:
##        labels
## 1 Accessories
## 2  Appliances
## 3         Art
##
## includes extended transaction information - examples:
##   transactionID
## 1  AE-2011-9160
## 2  AE-2013-1130
## 3  AE-2013-1530
```

```
#visually showing most frequent ittems
itemFrequencyPlot(arm_transactions,topN=20,type="absolute")
```




```
# find some initial rules
arm.rules <- apriori(arm_transactions, parameter=list(support=0.0005, conf=0.4,
target="rules"))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.4    0.1    1 none FALSE            TRUE       5   5e-04      1
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 12
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[17 item(s), 25035 transaction(s)] done [0.00s].
## sorting and recoding items ... [17 item(s)] done [0.00s].
## creating transaction tree ... done [0.01s].
## checking subsets of size 1 2 3 4 5 done [0.00s].
## writing ... [162 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```r
#takes a lot of trial-error to arrive at this final results
summary(arm.rules)
```

```
## set of 162 rules
##
## rule length distribution (lhs + rhs):sizes
##   4   5
## 151  11
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   4.000   4.000   4.000   4.068   4.000   5.000
##
## summary of quality measures:
##     support            confidence        coverage              lift
##  Min.   :0.0005193   Min.   :0.4000   Min.   :0.0007989   Min.   :1.857
##  1st Qu.:0.0005592   1st Qu.:0.4093   1st Qu.:0.0012782   1st Qu.:1.974
##  Median :0.0007190   Median :0.4286   Median :0.0016777   Median :2.217
##  Mean   :0.0007927   Mean   :0.4469   Mean   :0.0018054   Mean   :2.273
##  3rd Qu.:0.0009187   3rd Qu.:0.4607   3rd Qu.:0.0021470   3rd Qu.:2.431
##  Max.   :0.0019972   Max.   :0.7000   Max.   :0.0048732   Max.   :3.885
##      count
##  Min.   :13.00
##  1st Qu.:14.00
##  Median :18.00
##  Mean   :19.85
##  3rd Qu.:23.00
##  Max.   :50.00
##
## mining info:
##              data ntransactions support confidence
##  arm_transactions         25035   5e-04        0.4
##                                                                                                  call
##  apriori(data = arm_transactions, parameter = list(support = 5e-04, conf = 0.4, target = "rules"))
```

```r
#f_basket <- as(arm.rules,"data.frame")
top_lift<-head(arm.rules, n=10, by= "lift")
inspect(top_lift)
```

```
##      lhs                                          rhs           support
## [1]  {Binders, Labels, Phones, Storage}      => {Accessories} 0.0005192730
## [2]  {Appliances, Machines, Storage}         => {Art}         0.0005192730
## [3]  {Appliances, Art, Binders, Furnishings} => {Storage}     0.0005592171
## [4]  {Accessories, Binders, Labels, Storage} => {Phones}      0.0005192730
## [5]  {Fasteners, Labels, Machines}           => {Art}         0.0006391053
## [6]  {Appliances, Art, Furnishings, Storage} => {Binders}     0.0005592171
## [7]  {Art, Binders, Paper, Storage}          => {Chairs}      0.0005592171
## [8]  {Labels, Machines, Paper}               => {Art}         0.0005192730
## [9]  {Accessories, Labels, Phones, Storage}  => {Binders}     0.0005192730
## [10] {Accessories, Binders, Labels, Phones}  => {Storage}     0.0005192730
##      confidence coverage     lift     count
## [1]  0.4482759  0.0011583783 3.884592 13
## [2]  0.6190476  0.0008388256 3.549670 13
```

```
## [3]  0.6086957  0.0009187138 3.360983 14
## [4]  0.4193548  0.0012382664 3.350957 13
## [5]  0.5714286  0.0011184342 3.276618 16
## [6]  0.7000000  0.0007988816 3.250093 14
## [7]  0.4000000  0.0013980427 3.142140 14
## [8]  0.5416667  0.0009586579 3.105961 13
## [9]  0.6500000  0.0007988816 3.017943 13
## [10] 0.5416667  0.0009586579 2.990875 13
```

```
top_conf<-head(arm.rules, n=10, by= "confidence")
inspect(top_conf)
```

```
##      lhs                                          rhs       support
## [1]  {Appliances, Art, Furnishings, Storage} => {Binders} 0.0005592171
## [2]  {Accessories, Labels, Phones, Storage}  => {Binders} 0.0005192730
## [3]  {Appliances, Machines, Storage}         => {Art}     0.0005192730
## [4]  {Appliances, Art, Binders, Furnishings} => {Storage} 0.0005592171
## [5]  {Appliances, Paper, Storage}            => {Binders} 0.0012382664
## [6]  {Appliances, Furnishings, Storage}      => {Binders} 0.0011184342
## [7]  {Fasteners, Labels, Machines}           => {Art}     0.0006391053
## [8]  {Appliances, Fasteners, Paper}          => {Binders} 0.0007189934
## [9]  {Art, Chairs, Paper, Storage}           => {Binders} 0.0005592171
## [10] {Appliances, Fasteners, Phones}         => {Binders} 0.0006391053
##      confidence coverage     lift     count
## [1]  0.7000000  0.0007988816 3.250093 14
## [2]  0.6500000  0.0007988816 3.017943 13
## [3]  0.6190476  0.0008388256 3.549670 13
## [4]  0.6086957  0.0009187138 3.360983 14
## [5]  0.6078431  0.0020371480 2.822209 31
## [6]  0.5957447  0.0018773717 2.766036 28
## [7]  0.5714286  0.0011184342 3.276618 16
## [8]  0.5625000  0.0012782105 2.611682 18
## [9]  0.5600000  0.0009986020 2.600074 14
## [10] 0.5517241  0.0011583783 2.561649 16
```

```
top_lift <- head(arm.rules, n = 10, by = "lift")

inspect(top_lift)
```

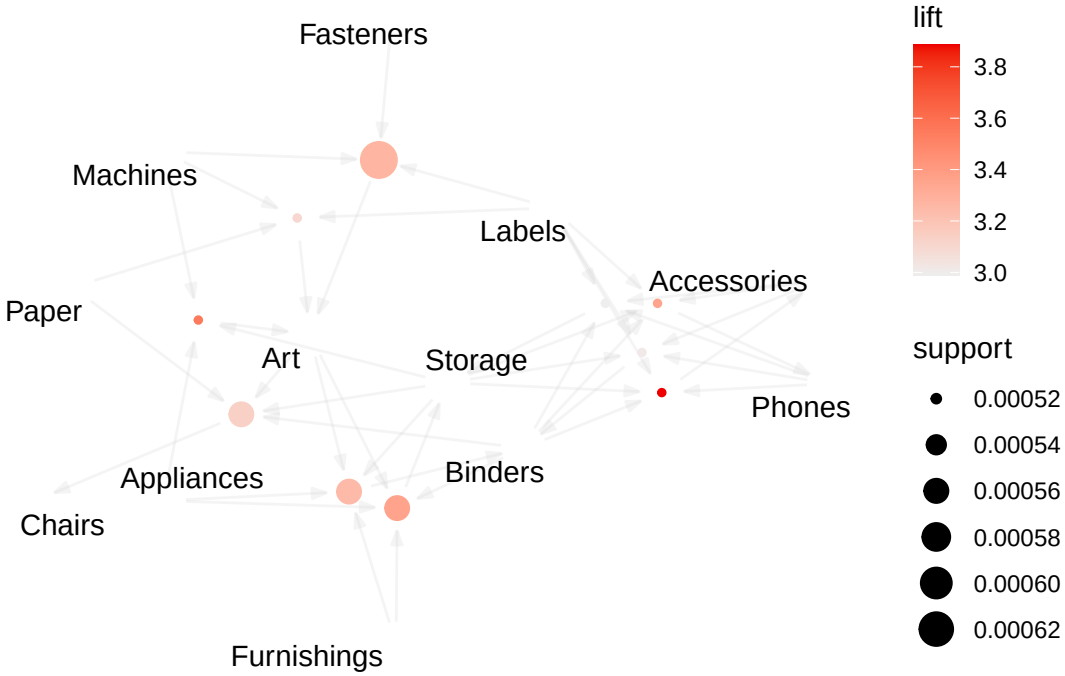
```
##      lhs                                          rhs           support
## [1]  {Binders, Labels, Phones, Storage}      => {Accessories} 0.0005192730
## [2]  {Appliances, Machines, Storage}         => {Art}         0.0005192730
## [3]  {Appliances, Art, Binders, Furnishings} => {Storage}     0.0005592171
## [4]  {Accessories, Binders, Labels, Storage} => {Phones}      0.0005192730
## [5]  {Fasteners, Labels, Machines}           => {Art}         0.0006391053
## [6]  {Appliances, Art, Furnishings, Storage} => {Binders}     0.0005592171
## [7]  {Art, Binders, Paper, Storage}          => {Chairs}      0.0005592171
## [8]  {Labels, Machines, Paper}               => {Art}         0.0005192730
## [9]  {Accessories, Labels, Phones, Storage}  => {Binders}     0.0005192730
## [10] {Accessories, Binders, Labels, Phones}  => {Storage}     0.0005192730
##      confidence coverage     lift     count
## [1]  0.4482759  0.0011583783 3.884592 13
## [2]  0.6190476  0.0008388256 3.549670 13
```

```
## [3]  0.6086957  0.0009187138 3.360983 14
## [4]  0.4193548  0.0012382664 3.350957 13
## [5]  0.5714286  0.0011184342 3.276618 16
## [6]  0.7000000  0.0007988816 3.250093 14
## [7]  0.4000000  0.0013980427 3.142140 14
## [8]  0.5416667  0.0009586579 3.105961 13
## [9]  0.6500000  0.0007988816 3.017943 13
## [10] 0.5416667  0.0009586579 2.990875 13
```

```
group.hi<-head(arm.rules,by="lift", 10)
plot(group.hi,method="graph", control=list(type="items"))
```

```
## Warning: Unknown control parameters: type

## Available control parameters (with default values):
## layout    =  stress
## circular  =  FALSE
## ggraphdots    =  NULL
## edges     =  <environment>
## nodes     =  <environment>
## nodetext  =  <environment>
## colors    =  c("#EE0000FF", "#EEEEEEFF")
## engine    =  ggplot2
## max   =  100
## verbose   =  FALSE
```



```r
inspect(group.hi)
```

```
##      lhs                                          rhs           support
## [1]  {Binders, Labels, Phones, Storage}      => {Accessories} 0.0005192730
## [2]  {Appliances, Machines, Storage}         => {Art}         0.0005192730
## [3]  {Appliances, Art, Binders, Furnishings} => {Storage}     0.0005592171
## [4]  {Accessories, Binders, Labels, Storage} => {Phones}      0.0005192730
## [5]  {Fasteners, Labels, Machines}           => {Art}         0.0006391053
## [6]  {Appliances, Art, Furnishings, Storage} => {Binders}     0.0005592171
## [7]  {Art, Binders, Paper, Storage}          => {Chairs}      0.0005592171
## [8]  {Labels, Machines, Paper}               => {Art}         0.0005192730
## [9]  {Accessories, Labels, Phones, Storage}  => {Binders}     0.0005192730
## [10] {Accessories, Binders, Labels, Phones}  => {Storage}     0.0005192730
##      confidence coverage     lift     count
## [1]  0.4482759  0.0011583783 3.884592 13
## [2]  0.6190476  0.0008388256 3.549670 13
## [3]  0.6086957  0.0009187138 3.360983 14
## [4]  0.4193548  0.0012382664 3.350957 13
## [5]  0.5714286  0.0011184342 3.276618 16
## [6]  0.7000000  0.0007988816 3.250093 14
## [7]  0.4000000  0.0013980427 3.142140 14
## [8]  0.5416667  0.0009586579 3.105961 13
## [9]  0.6500000  0.0007988816 3.017943 13
## [10] 0.5416667  0.0009586579 2.990875 13
```

```r
#our code applies market basket analysis to retail transaction data,
#uncovering patterns in customer purchasing behavior. By focusing on the
#Order.ID and Sub.Category columns, the dataset was prepared for analysis as a
#sparse transactions object, capturing 25,035 unique orders across 17 product
#subcategories. Frequent items such as Binders, Storage, and Art highlighted
#their importance, setting the stage for association rule mining.

#This code also emphasizes the practical implications of the findings.
#The discovered rules can inform cross-selling strategies, such as bundling
#Storage and Art with Binders to increase basket sizes. Moreover, the insights
#can be used to optimize product placement in stores or recommend complementary
#products in e-commerce settings. For instance, customers frequently purchasing
#Appliances and Storage are likely to also buy Art, providing an opportunity to
#tailor promotions or recommendations.

#The ability to mine these associations demonstrates the power of leveraging
#data for strategic decision-making. By focusing on product subcategories, the
#analysis remains adaptable to various contexts, whether in physical retail or
#digital commerce. The strong associations revealed in the rules highlight not
#just customer preferences but also potential avenues for improving operational
#efficiency, such as inventory management and demand forecasting.

#To sum up, this code offers a solid approach to identifying and understanding
#complex item relationships in transactional data. By combining thorough data
#preparation, effective algorithms, and clear visualizations, it delivers
#results that are both practical and insightful. These findings highlight how
#market basket analysis can be a powerful tool for analyzing customer behavior
#and driving smarter business strategies.
```

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Ctrl+Alt+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Ctrl+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.