Product Recommendation System using Cosine Similarity

Under the guidance of Dr.Cathy Durso

By:
- Aakarsh Sagar
- Ajay Rawtani

Probability and Statistics [4442-1]

Contents

Introduction

A product recommendation system is an advanced technology that analyzes user data and behaviors to provide personalized suggestions for products or services. By leveraging machine learning algorithms, these systems can accurately predict user preferences and offer relevant recommendations, thereby enhancing the user experience.

For small businesses, product recommendation systems can be particularly beneficial. Firstly, they enable small businesses to compete with larger enterprises by delivering tailored suggestions to customers, creating a more personalized shopping experience. This increases customer satisfaction and loyalty, leading to repeat purchases and positive word-of-mouth.

Furthermore, product recommendation systems contribute to revenue growth through upselling and cross-selling. By suggesting complementary or higher-priced items based on user preferences or previous purchases, businesses can encourage customers to spend more. This strategy effectively maximizes the value of each transaction and boosts overall revenue.

Additionally, these systems allow small businesses to make data-driven decisions by gathering insights into customer preferences and market trends. This information can be used to optimize product offerings, marketing strategies, and inventory management, leading to improved operational efficiency and profitability.

Overall, product recommendation systems empower small businesses to leverage customer data effectively, drive sales, and foster long-term customer relationships.

Data Source

The dataset chosen for this project is a primary data set sourced from an actual brick-and-mortar fabric store located in Hyderabad, India. It has been carefully selected to ensure its relevance and applicability to our project objectives. The data source has been made available alongside the project upload, facilitating further analysis and exploration.

Overview of Dataset

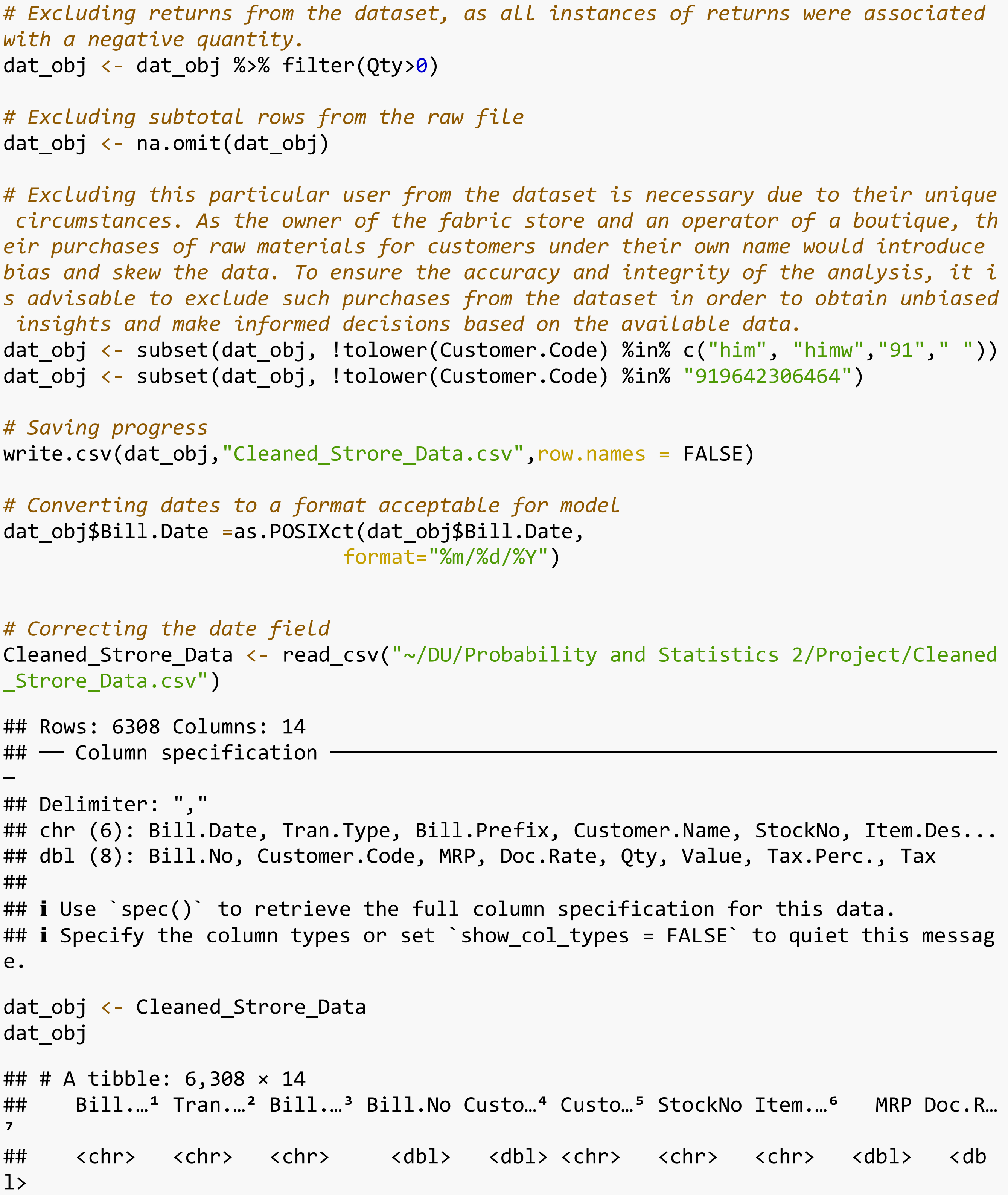
The dataset consists of 14 variables and 8897 row entries. The data spans from April 2020 to March 2022, aligning with the financial year of April to March of the following year. Prior to analysis and model implementation, the data will require preprocessing to handle missing values and ensure readiness for further analysis. It is important to note that “Customer.Code” uniquely identifies a customer.

```
dat_obj <- read.csv("Store_Data.csv")
head(dat_obj)
```

```
##                    Bill.Date Tran.Type Bill.Prefix Bill.No Customer.Code
## 1                 01/01/2021     Sales         S20     969  919701332445
## 2                 01/01/2021     Sales         S20     969  919701332445
## 3                 01/01/2021     Sales         S20     969  919701332445
## 4                 01/01/2021     Sales         S20     970  919347983667
## 5                *Sub Total*   *Sales*                  NA
## 6 *Sub Total* - *01/01/2021*                            NA
##   Customer.Name   StockNo                  Item.Description  MRP Doc.Rate   Qty
## 1       Saritha  819ST172     DIG RJALL5079B CHANDERI LUREX  325      400  2.00
## 2       Saritha   120ST57 DIG MEALL1744 ORGANZA SATIN NYLON  280      350  2.00
## 3       Saritha 417ST3506                   DYED TOUCH FEEL  145      145  1.75
## 4        shilpa  1220ST98   DYED EMB. GT-75006 PURE ORGANZA 1095     1095  5.50
## 5                                                             NA       NA 11.25
## 6                                                             NA       NA 11.25
##   Value Tax.Perc.    Tax
## 1   800         5  38.10
## 2   700         5  33.33
## 3   254         5  12.08
## 4  6023         5 286.79
## 5  7777        NA 370.30
## 6  7777        NA 370.30
```

Data Cleaning

Data preparation was conducted using the R programming language, as depicted in the code below. To ensure clarity and comprehension, detailed comments have been provided within the code.



```
# Excluding returns from the dataset, as all instances of returns were associated
with a negative quantity.
dat_obj <- dat_obj %>% filter(Qty>0)

# Excluding subtotal rows from the raw file
dat_obj <- na.omit(dat_obj)

# Excluding this particular user from the dataset is necessary due to their unique
 circumstances. As the owner of the fabric store and an operator of a boutique, th
eir purchases of raw materials for customers under their own name would introduce
bias and skew the data. To ensure the accuracy and integrity of the analysis, it i
s advisable to exclude such purchases from the dataset in order to obtain unbiased
 insights and make informed decisions based on the available data.
dat_obj <- subset(dat_obj, !tolower(Customer.Code) %in% c("him", "himw","91"," "))
dat_obj <- subset(dat_obj, !tolower(Customer.Code) %in% "919642306464")

# Saving progress
write.csv(dat_obj,"Cleaned_Strore_Data.csv",row.names = FALSE)

# Converting dates to a format acceptable for model
dat_obj$Bill.Date =as.POSIXct(dat_obj$Bill.Date,
                              format="%m/%d/%Y")


# Correcting the date field
Cleaned_Strore_Data <- read_csv("~/DU/Probability and Statistics 2/Project/Cleaned
_Strore_Data.csv")

## Rows: 6308 Columns: 14
## ── Column specification ────────────────────────────────────────────────────────
─
## Delimiter: ","
## chr (6): Bill.Date, Tran.Type, Bill.Prefix, Customer.Name, StockNo, Item.Des...
## dbl (8): Bill.No, Customer.Code, MRP, Doc.Rate, Qty, Value, Tax.Perc., Tax
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this messag
e.

dat_obj <- Cleaned_Strore_Data
dat_obj

## # A tibble: 6,308 × 14
##    Bill.…¹ Tran.…² Bill.…³ Bill.No Custo…⁴ Custo…⁵ StockNo Item.…⁶   MRP Doc.R…
⁷
##    <chr>   <chr>   <chr>     <dbl>   <dbl> <chr>   <chr>   <chr>   <dbl>   <db
l>
```

```
##  1 01/01/… Sales   S20         969 9.20e11 Saritha 819ST1… DIG RJ…   325     40
0
##  2 01/01/… Sales   S20         969 9.20e11 Saritha 120ST57 DIG ME…   280     35
0
##  3 01/01/… Sales   S20         969 9.20e11 Saritha 417ST3… DYED T…   145     14
5
##  4 01/01/… Sales   S20         970 9.19e11 shilpa  1220ST… DYED E…  1095    109
5
##  5 01/01/… Sales   S21         871 9.20e11 SANTOS… 112ST1… DYED N…    89      8
9
##  6 01/02/… Sales   S20        1235 9.19e11 amaral… 1220ST… DIG SD…   550     45
0
##  7 01/02/… Sales   S20        1235 9.19e11 amaral… 1220ST… DIG NK…   550     45
0
##  8 01/02/… Sales   S20        1236 9.19e11 aruna … 1120ST… RFD RA…   210     21
0
##  9 01/02/… Sales   S20        1236 9.19e11 aruna … 1219ST… DUPATT…  1105    110
5
## 10 01/02/… Sales   S20        1236 9.19e11 aruna … 1220ST… EMB. O…   450     45
0
## # … with 6,298 more rows, 4 more variables: Qty <dbl>, Value <dbl>,
## #   Tax.Perc. <dbl>, Tax <dbl>, and abbreviated variable names ¹Bill.Date,
## #   ²Tran.Type, ³Bill.Prefix, ⁴Customer.Code, ⁵Customer.Name,
## #   ⁶Item.Description, ⁷Doc.Rate

dat_obj$Bill.Date <- as.Date(dat_obj$Bill.Date, format = "%d/%m/%Y")
dat_obj

## # A tibble: 6,308 × 14
##    Bill.Date  Tran.Type Bill.Pre…¹ Bill.No Custo…² Custo…³ StockNo Item.…⁴    MR
P
##    <date>     <chr>     <chr>        <dbl>   <dbl> <chr>   <chr>   <chr>    <db
l>
##  1 2021-01-01 Sales     S20            969 9.20e11 Saritha 819ST1… DIG RJ…    32
5
##  2 2021-01-01 Sales     S20            969 9.20e11 Saritha 120ST57 DIG ME…    28
0
##  3 2021-01-01 Sales     S20            969 9.20e11 Saritha 417ST3… DYED T…    14
5
##  4 2021-01-01 Sales     S20            970 9.19e11 shilpa  1220ST… DYED E…   109
5
##  5 2022-01-01 Sales     S21            871 9.20e11 SANTOS… 112ST1… DYED N…     8
9
##  6 2021-02-01 Sales     S20           1235 9.19e11 amaral… 1220ST… DIG SD…    55
0
##  7 2021-02-01 Sales     S20           1235 9.19e11 amaral… 1220ST… DIG NK…    55
0
##  8 2021-02-01 Sales     S20           1236 9.19e11 aruna … 1120ST… RFD RA…    21
0
##  9 2021-02-01 Sales     S20           1236 9.19e11 aruna … 1219ST… DUPATT…   110
5
## 10 2021-02-01 Sales     S20           1236 9.19e11 aruna … 1220ST… EMB. O…    45
0
```

```
## # … with 6,298 more rows, 5 more variables: Doc.Rate <dbl>, Qty <dbl>,
## #   Value <dbl>, Tax.Perc. <dbl>, Tax <dbl>, and abbreviated variable names
## #   ¹Bill.Prefix, ²Customer.Code, ³Customer.Name, ⁴Item.Description
```

```
dat_obj$Bill.Date <- format(dat_obj$Bill.Date, "%Y-%m-%d")

# In order to meet the requirement, it is necessary to create a new field called "
bill.item" to serve as a unique identifier for each bill. The existing bill number
s and years were found to be insufficient in satisfying this requirement:
dat_obj$Bill.Item <- paste0(as.character(dat_obj$Bill.No),"_",as.character(dat_obj
$Bill.Prefix))
dat_obj
```

```
## # A tibble: 6,308 × 15
##    Bill.…¹ Tran.…² Bill.…³ Bill.No Custo…⁴ Custo…⁵ StockNo Item.…⁶   MRP Doc.R…
⁷
##    <chr>   <chr>   <chr>     <dbl>   <dbl> <chr>   <chr>   <chr>   <dbl>   <db
l>
##  1 2021-0… Sales   S20         969 9.20e11 Saritha 819ST1… DIG RJ…   325     40
0
##  2 2021-0… Sales   S20         969 9.20e11 Saritha 120ST57 DIG ME…   280     35
0
##  3 2021-0… Sales   S20         969 9.20e11 Saritha 417ST3… DYED T…   145     14
5
##  4 2021-0… Sales   S20         970 9.19e11 shilpa  1220ST… DYED E…  1095    109
5
##  5 2022-0… Sales   S21         871 9.20e11 SANTOS… 112ST1… DYED N…    89      8
9
##  6 2021-0… Sales   S20        1235 9.19e11 amaral… 1220ST… DIG SD…   550     45
0
##  7 2021-0… Sales   S20        1235 9.19e11 amaral… 1220ST… DIG NK…   550     45
0
##  8 2021-0… Sales   S20        1236 9.19e11 aruna … 1120ST… RFD RA…   210     21
0
##  9 2021-0… Sales   S20        1236 9.19e11 aruna … 1219ST… DUPATT…  1105    110
5
## 10 2021-0… Sales   S20        1236 9.19e11 aruna … 1220ST… EMB. O…   450     45
0
## # … with 6,298 more rows, 5 more variables: Qty <dbl>, Value <dbl>,
## #   Tax.Perc. <dbl>, Tax <dbl>, Bill.Item <chr>, and abbreviated variable names
## #   ¹Bill.Date, ²Tran.Type, ³Bill.Prefix, ⁴Customer.Code, ⁵Customer.Name,
## #   ⁶Item.Description, ⁷Doc.Rate
```

```
# Saving document
write.csv(dat_obj, "Updated_Store_Data_Cleaned.csv", row.names = FALSE)
```

Research Question

Exploratory Data Analysis:

Which products are the best sellers in the store?

Are there any noticeable revenue trends, and is there a cyclical pattern?

What is the distribution between unique and repeat customers?

Can cosine similarity be utilized to recommend additional items based on user similarity?

Can cosine similarity be employed to recommend additional items based on item similarity?

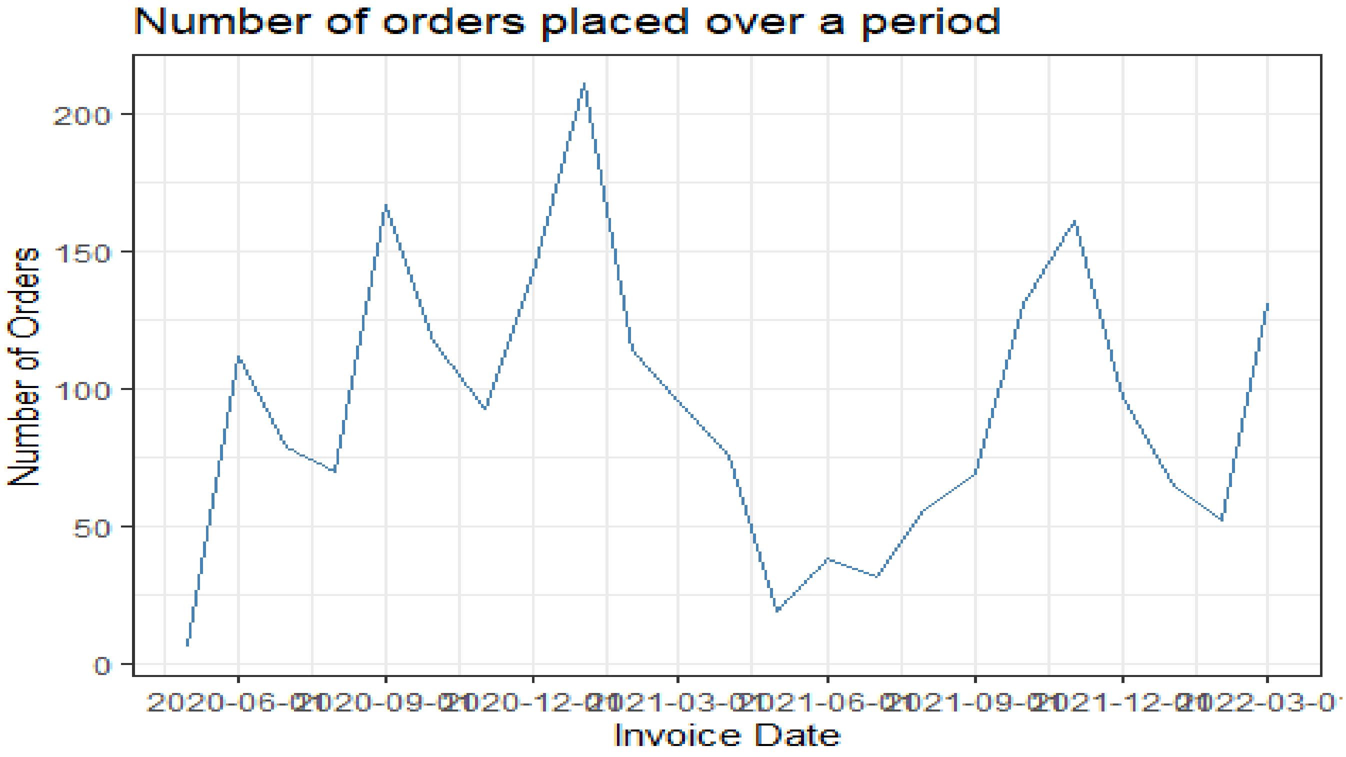
Application and Interpretation

Exploratory Data Analysis (EDA)

Order-wise analysis

Plotting some of the data to gain insights before performing any sort of analysis on the product data

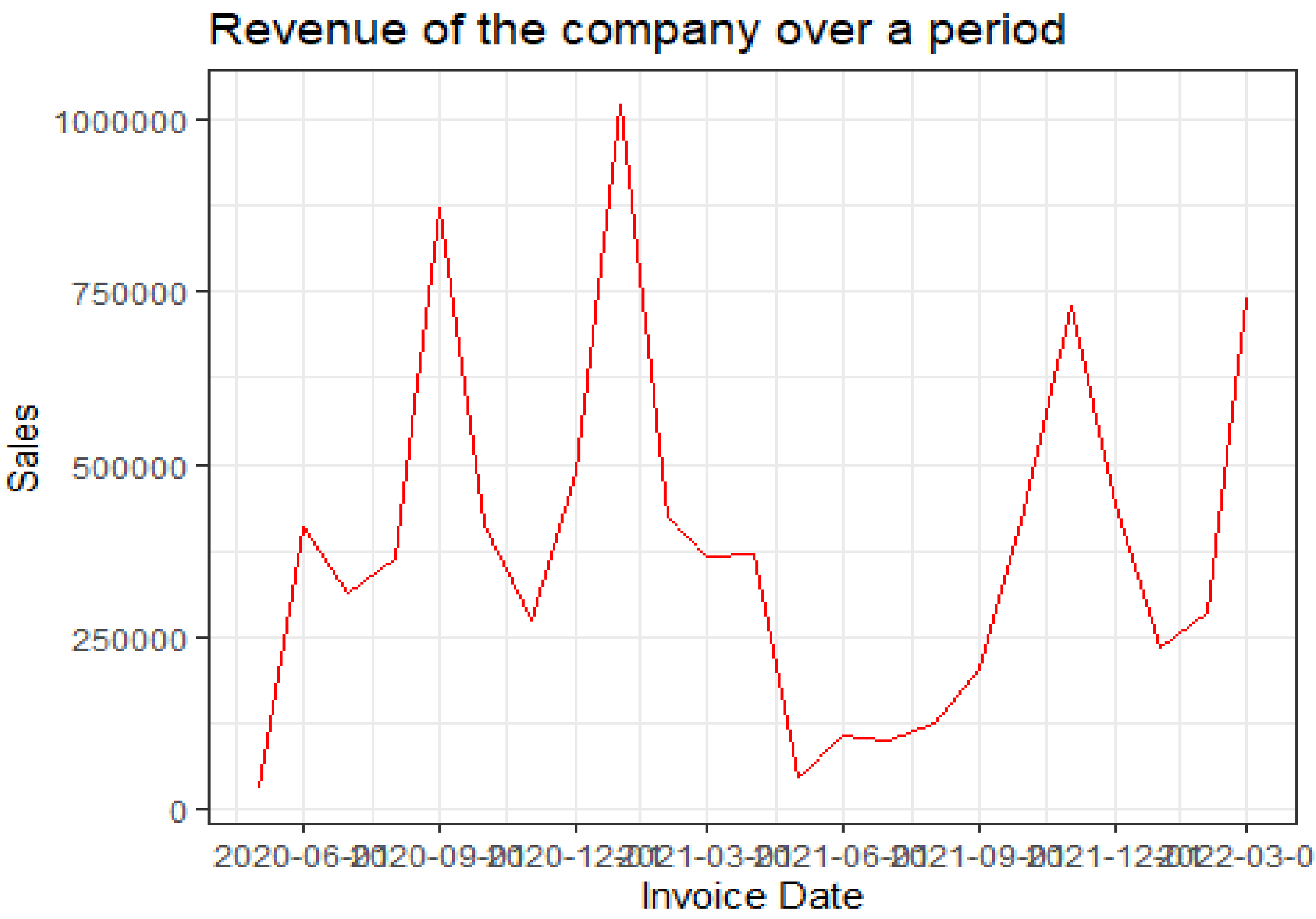
```r
# Reading file
dat_obj <- read.csv("Updated_Store_Data_Cleaned.csv")
dat_obj$Bill.Date <- as.Date(dat_obj$Bill.Date, format = "%Y-%m-%d")

# The monthly order volume trend over time
total_invoices <- dat_obj %>%
  group_by(Bill.Date=floor_date(Bill.Date, "month")) %>%
  summarise(number_of_orders=n_distinct(Bill.Item))

ggplot(total_invoices, aes(x = Bill.Date, y = number_of_orders)) +
  geom_line(color = "steelblue") +
  labs(x = "Invoice Date", y = "Number of Orders") +
  ggtitle("Number of orders placed over a period") +
  theme_bw() + scale_x_date(date_breaks = "3 month")
```



This visual representation depicts a time-series plot showcasing dates on the x-axis and the corresponding daily order count on the y-axis. Upon analyzing the plot, it is evident that the order count reaches its highest point shortly after September, coinciding with the wedding season observed in India from January to March. During this period, it is customary for individuals attending weddings to adorn themselves in intricately tailored garments.

Additionally, a substantial decline is observed between May 2021 and September 2021, which can be attributed to the second wave of the COVID-19 pandemic. During this period, the store operated for only a limited number of days each week, resulting in a significant decrease in order activity.

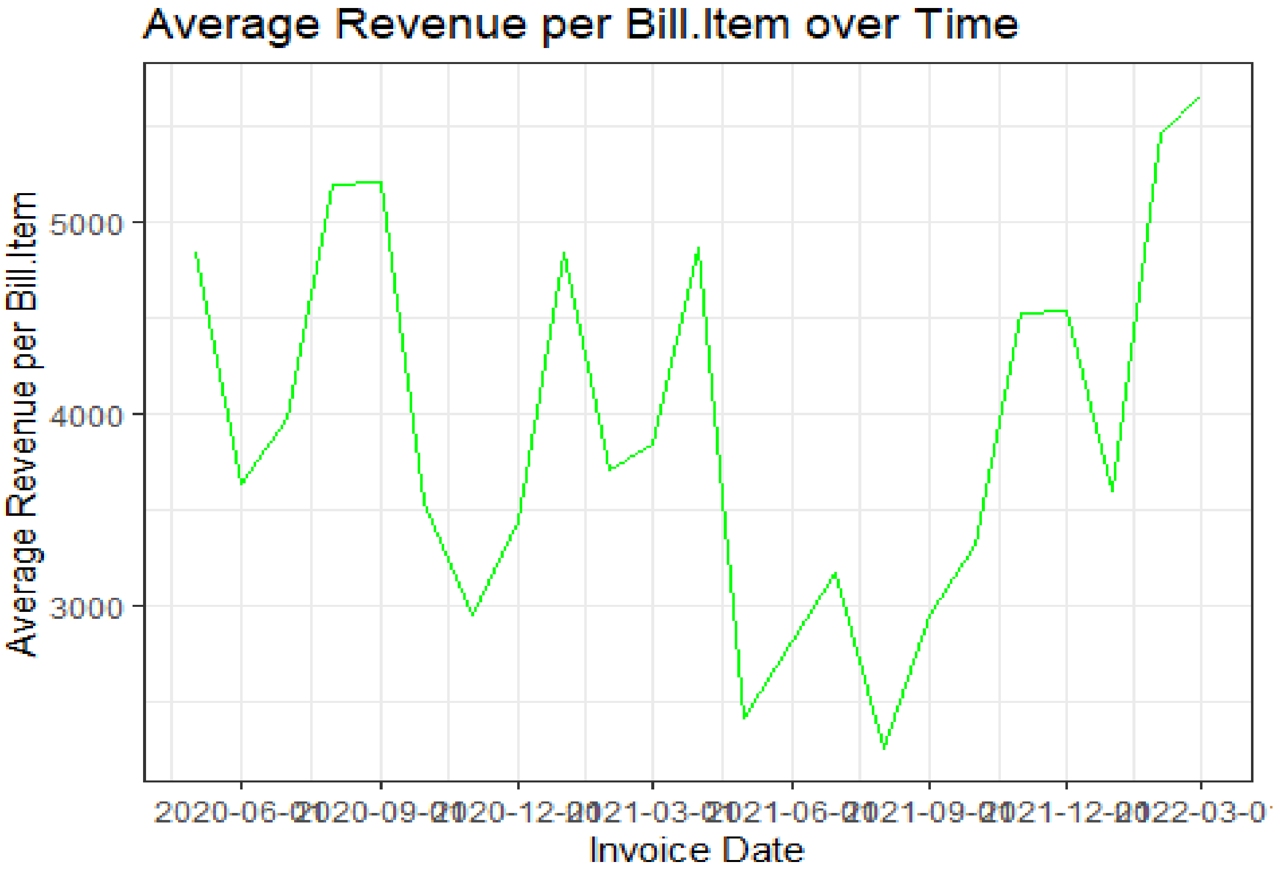
```
# The monthly revenue trend over time
revenue_over_time <- dat_obj %>%
  group_by(Bill.Date=floor_date(Bill.Date, "month")) %>%
  summarise(Sales=sum(Value))

ggplot(revenue_over_time, aes(x= Bill.Date, y=Sales)) + geom_line(color = "red")+
labs(x= "Invoice Date", y = "Sales") + ggtitle("Revenue of the company over a peri
od")+theme_bw() + scale_x_date(date_breaks = "3 month")
```



This visual representation depicts a time-series plot showcasing dates on the x-axis and the corresponding daily revenue on the y-axis. Upon analyzing the plot, it is evident that the revenue reaches its highest point shortly after September, coinciding with the wedding season observed in India from January to March. During this period, it is customary for individuals attending weddings to adorn themselves in intricately tailored garments.

Additionally, a substantial decline is observed between May 2021 and September 2021, which can be attributed to the second wave of the COVID-19 pandemic. During this period, the store operated for only a limited number of days each week, resulting in a significant decrease in order activity.

```r
# Average Revenue per Bill over time (important metric from business POV)
average_revenue <- dat_obj %>%
  group_by(Bill.Date=floor_date(Bill.Date, "month")) %>%
  summarise(average_revenue = sum(Value) / n_distinct(Bill.Item))

ggplot(average_revenue, aes(x = Bill.Date, y = average_revenue)) +
  geom_line(color = "green") +
  labs(x = "Invoice Date", y = "Average Revenue per Bill.Item") +
  ggtitle("Average Revenue per Bill.Item over Time") +
  theme_bw() +
  scale_x_date(date_breaks = "3 month")
```



Maximizing revenue for each bill generated is a crucial metric for any business, as it enables the organization to focus on a specific user group and adapt its strategy, accordingly, ensuring long-term success.

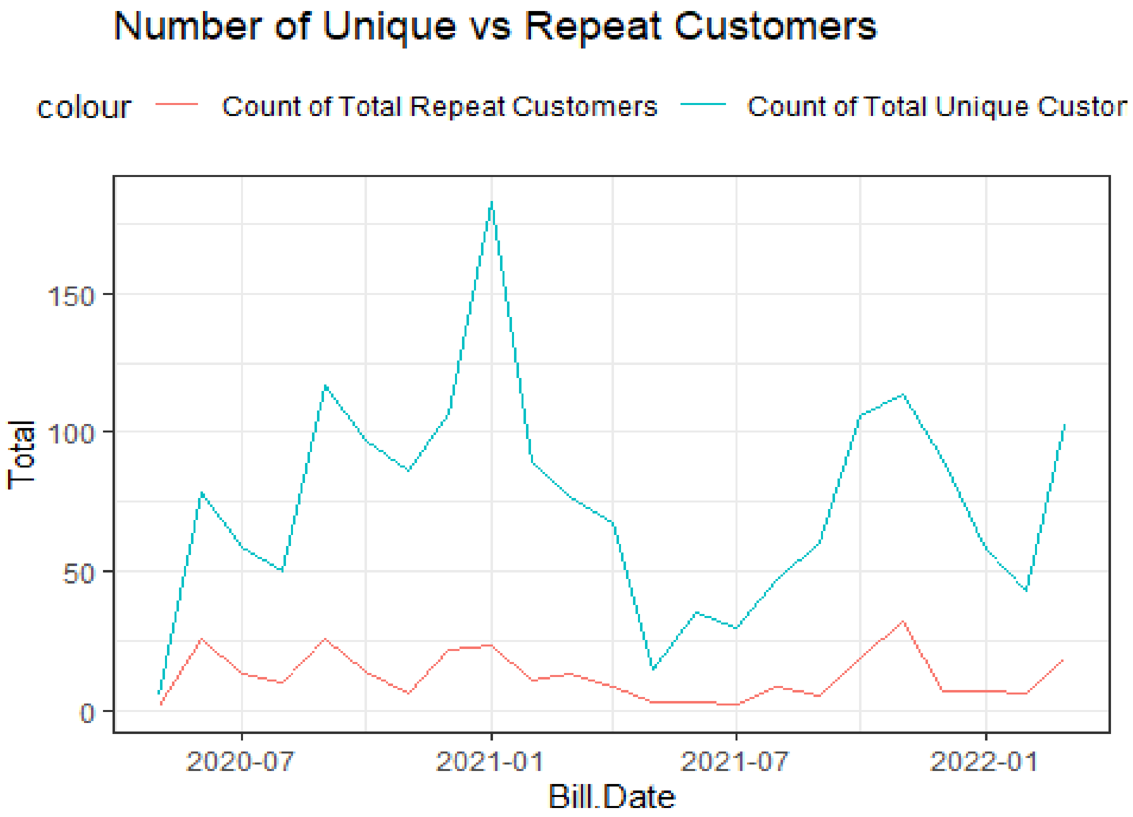
Upon examining the graph, excluding the period affected by the COVID-19 pandemic, it is evident that the trend indicates an Average Revenue Per User (ARPU) slightly above ₹4000 (~$50). However, a concerning observation emerges from the data: the ARPU has remained constant over a three-year timeframe, despite inflation, rising input costs, and general increases in wages year after year. Ideally, the ARPU should have shown a gradual increase over time.

Repeat Customers

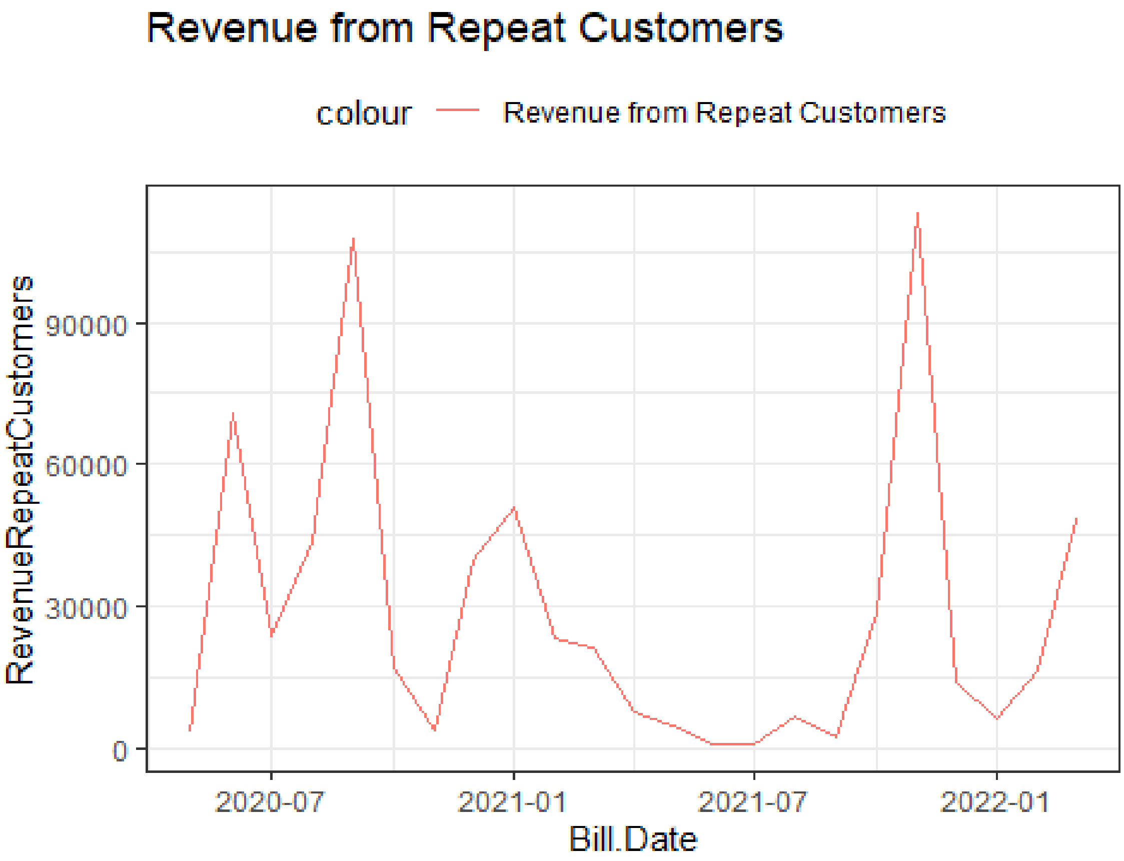
The 80-20 Rule, also known as the Pareto Principle, is a well-established principle in business that states that 80% of revenue is typically generated by 20% of customers. This principle highlights the importance of repeat customers for the overall success of a business. A high number of repeat customers is essential for a business to maintain sustainability and long-term growth.

```
#aggregating data so that one row represents one purchase order
Invoice <- dat_obj %>%
  group_by(Bill.Date, Bill.Item) %>%
  summarise(total_sales = sum(Value), total_quantity = sum(Qty),Customer.Code = ma
x(Customer.Code))

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

#aggregating data into months
InvoiceCustomer <-
  Invoice %>%
  group_by(Bill.Date = floor_date(Bill.Date, "month"), Customer.Code) %>%
  summarise(Count = n_distinct(Bill.Item), Sales = sum(total_sales))

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

InvoiceCustomer

## # A tibble: 1,716 × 4
## # Groups:   Bill.Date [23]
##    Bill.Date  Customer.Code Count Sales
##    <date>             <dbl> <int> <dbl>
##  1 2020-05-01  918019019602     1   653
##  2 2020-05-01  919092436348     1  4910
##  3 2020-05-01  919491805526     1  5233
##  4 2020-05-01  919703323131     2 15190
```

```
write.csv(InvoiceCustomer,"InvoiceCustomer.csv", row.names = FALSE)

# Repeat Customers are count of unique customer.code grouped by date
RepeatCustomers <- InvoiceCustomer %>%
  filter(Count > 1)

RepeatCustomers <- RepeatCustomers %>%
  group_by(Bill.Date) %>%
  summarize(Count=n_distinct(Customer.Code), Sales=sum(Sales))
RepeatCustomers

## # A tibble: 23 × 3
##    Bill.Date  Count  Sales
##    <date>     <int>  <dbl>
##  1 2020-05-01     1  15190
##  2 2020-06-01    26 212474
##  3 2020-07-01    13 108029
##  4 2020-08-01    10 217034

#total number of monthly customers
UniqueCustomers <- dat_obj %>%
  group_by(Bill.Date=floor_date(Bill.Date, "month")) %>%
  summarise(Count=n_distinct(Customer.Code))

#find the percentage of monthly revenue that are attributed to the repeat customer
s
RepeatCustomers$Perc <- RepeatCustomers$Sales/revenue_over_time$Sales*100.0

#append unique customers
RepeatCustomers$Total <- UniqueCustomers$Count


ggplot(RepeatCustomers) +
  geom_line(aes(x=Bill.Date, y=Total, color="Count of Total Unique Customers"), st
at="identity") +
  geom_line(aes(x=Bill.Date, y=Count, color="Count of Total Repeat Customers"), st
at="identity") +
  # geom_bar(aes(x=Bill.Date, y=Perc*20, fill="Percentage of Revenue"), stat="iden
tity", alpha=0.5) +
  # scale_color_manual(values=c("orange", "navy"), labels=c("Total Unique Customer
s", "Repeat Customers")) +
  # scale_fill_manual(values="gray", labels="Revenue from repeat customers") +
  # scale_y_continuous(sec.axis = sec_axis(~./20, name="Percentage (%)")) +
  labs(title="Number of Unique vs Repeat Customers") +
  theme_bw() +
  theme(legend.position="top")
```



```
# Revenue from repeat customer
RepeatCustomers$RevenueRepeatCustomers <- RepeatCustomers$Sales * RepeatCustomers
$Count / RepeatCustomers$Total

# Plot the line chart
ggplot(RepeatCustomers) +
  geom_line(aes(x = Bill.Date, y = RevenueRepeatCustomers, color = "Revenue from R
epeat Customers"), stat = "identity") +
  labs(title = "Revenue from Repeat Customers") +
  theme_bw() +
  theme(legend.position = "top")
```



Top sellers

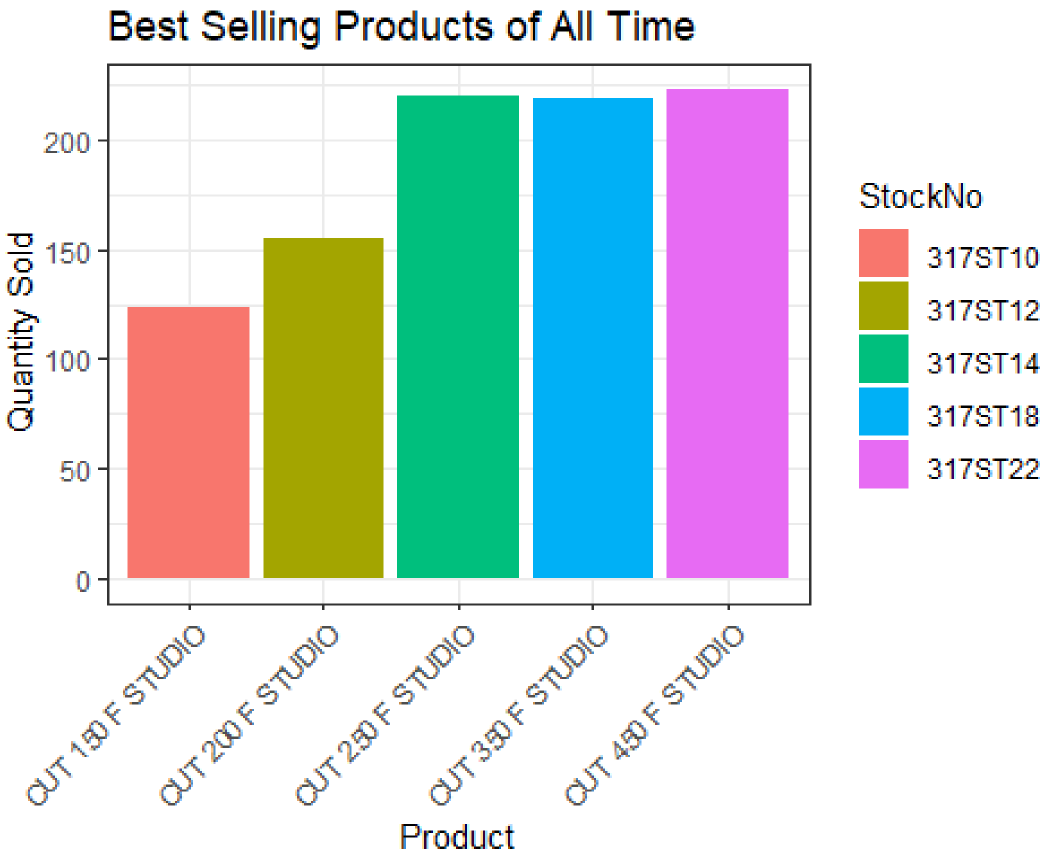
The top selling products in a store are an essential metric for evaluating success. This key indicator provides valuable insights into consumer preferences and market demand. By monitoring and analyzing the performance of these products, businesses can make informed decisions regarding inventory management, marketing strategies, and overall business growth.

```r
# Number of items sold for each product
pop.products <- dat_obj %>%
  group_by(StockNo) %>%
  summarise(Quantity = sum(Qty))

# Rank products based on total quantity sold
top.products <- pop.products %>%
  arrange(desc(Quantity)) %>%
  top_n(5)
```

```
## Selecting by Quantity
```

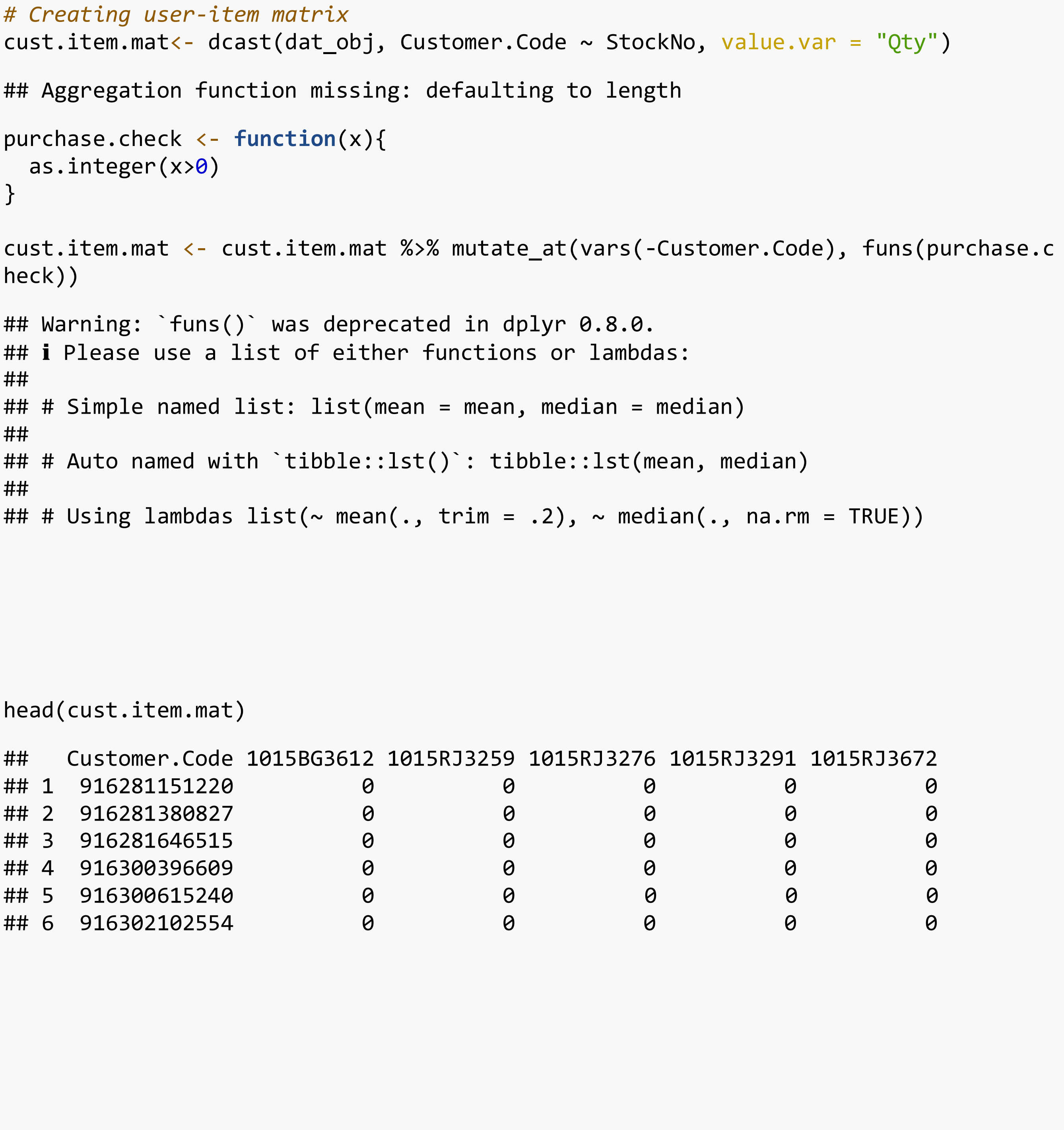
```r
# Retrieve stock names
stock_names <- dat_obj %>%
  filter(StockNo %in% top.products$StockNo) %>%
  distinct(StockNo, Item.Description)
```

```r
# Merge stock names with top.products
top.products <- top.products %>%
  left_join(stock_names, by = "StockNo")


# Plot the best selling products
ggplot(top.products, aes(x = Item.Description, y = Quantity, fill = StockNo)) +
  geom_bar(stat = "identity") +
  labs(title = "Best Selling Products of All Time",
       x = "Product",
       y = "Quantity Sold") +
  theme_bw() + theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Collaborative Filter model using Cosine Similarity

Collaborative filtering is a widely used recommendation technique that predicts user preferences by leveraging the behavior of similar users or items. By analyzing patterns and preferences of a community, it enables personalized recommendations, fosters discovery of new items, and enhances user engagement in various domains, including e-commerce and content streaming.



```
# Creating user-item matrix
cust.item.mat<- dcast(dat_obj, Customer.Code ~ StockNo, value.var = "Qty")

## Aggregation function missing: defaulting to length

purchase.check <- function(x){
  as.integer(x>0)
}

cust.item.mat <- cust.item.mat %>% mutate_at(vars(-Customer.Code), funs(purchase.c
heck))

## Warning: `funs()` was deprecated in dplyr 0.8.0.
## ℹ Please use a list of either functions or lambdas:
##
## # Simple named list: list(mean = mean, median = median)
##
## # Auto named with `tibble::lst()`: tibble::lst(mean, median)
##
## # Using lambdas list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
```

```
head(cust.item.mat)

##   Customer.Code 1015BG3612 1015RJ3259 1015RJ3276 1015RJ3291 1015RJ3672
## 1  916281151220          0          0          0          0          0
## 2  916281380827          0          0          0          0          0
## 3  916281646515          0          0          0          0          0
## 4  916300396609          0          0          0          0          0
## 5  916300615240          0          0          0          0          0
## 6  916302102554          0          0          0          0          0
```

User based collaborative filtering

User-based collaborative filtering is a popular recommendation technique that leverages the similarity between users' preferences to make personalized recommendations. By analyzing user behavior and preferences, it identifies similar users and suggests items that have been well-received by those with similar tastes, enhancing the overall user experience.

```
# Calculate the cosine values for user-user matrix
usercosine <- cosine(as.matrix(t(cust.item.mat[, 2:dim(cust.item.mat)[2]])))
colnames(usercosine) <- cust.item.mat$Customer.Code

# Reading user-number as input as customer number is customer code
#user_number <- readline("\nEnter a valid number: ")
user_number <- 916300615240
# Rank the most similar customers to our customer with Customer Code and picking t
op 5 similar users
Top5Similar <- cust.item.mat$Customer.Code[
  order(usercosine[, user_number], decreasing = TRUE)[2:6]]

## Warning in order(usercosine[, user_number], decreasing = TRUE): NAs introduced
## by coercion to integer range

# Similar Users print list
cat('\nTop 5 similar users:', Top5Similar)

##
## Top 5 similar users: 916281380827 916281646515 916300396609 916300615240 916302
102554

# Pick any one user at random from top 5 users and pull a list of products the sim
ilar user bought
sim_usr <- sample(Top5Similar, 1)
cat('\n Picking a similar user:', sim_usr)

##
##  Picking a similar user: 916300396609

boughtbyA <- cust.item.mat %>%
  filter(Customer.Code == user_number)
boughtbyA <- colnames(cust.item.mat)[which(boughtbyA !=0)]
cat('\nItem bought by select user', user_number,':\n')

##
## Item bought by select user 916300615240 :

boughtbyA[-1]

## [1] "317ST14" "317ST20" "317ST24"

# Let's find the descriptions of these items and return the description for refere
nce of store
boughtbyADescription <- unique(dat_obj[which(dat_obj$StockNo %in% boughtbyA),
     c("StockNo", "Item.Description")])
boughtbyADescription
```

```
##      StockNo Item.Description
## 3677 317ST14 CUT 250 F STUDIO
## 3679 317ST20 CUT 400 F STUDIO
## 3694 317ST24 CUT 500 F STUDIO

# Let's find what bought the B customer
boughtbyB <- cust.item.mat %>%
  filter(Customer.Code == sim_usr)
boughtbyB <- colnames(cust.item.mat)[which(boughtbyB !=0)]
cat('\nItem bought by similar user', user_number,':')

##
## Item bought by similar user 916300615240 :

boughtbyB[-1]

## [1] "1121ST381" "1121ST446"

# Let's find the items that the customer B didn't buy so we can recommend these it
ems to buy for B
RecommendToA <-setdiff(boughtbyB,boughtbyA)
cat("\nProduct Recommendations:", RecommendToA)

##
## Product Recommendations: 1121ST381 1121ST446

# Let's find the descriptions of these items
RecommendToADescription <- unique(
  dat_obj[which(dat_obj$StockNo %in% RecommendToA),
     c("StockNo", "Item.Description")])

RecommendToADescription <- RecommendToADescription[match(RecommendToA, RecommendTo
ADescription$StockNo),
]

# List of the items descriptions as a recommendation to B
cat('Product Recommended for purchase/marketing to A:')

## Product Recommended for purchase/marketing to A:

RecommendToADescription

##        StockNo            Item.Description
## 1346 1121ST381 DIG MEALL1913B COTTON SUPER
## 1340 1121ST446 DIG MEALL2246C COTTON SUPER
```

Item based collaborative filtering

Item-based collaborative filtering is a popular recommendation technique that analyzes the relationships between items based on user behavior. By identifying similar items, it recommends items to users based on their interactions with similar items, improving personalization and enhancing the accuracy and relevance of recommendations.

```r
itemcosine <- cosine(as.matrix(cust.item.mat[, 2:dim(cust.item.mat)[2]]))

# Read Stock Code
#usr_stockcode <- readline('Enter stock number: ')
usr_stockcode <- '1121ST469'

# Get Stock Code descriptions
StockDescriptions <- unique(
    dat_obj[which(dat_obj$StockNo %in% usr_stockcode), c("StockNo", "Item.Descript
ion")])
StockDescriptions

##        StockNo           Item.Description
## 2850 1121ST469 DIG SDALL2748 COTTON SUPER

# Find top5 most similar products to the product with Stock Code
Top5SimilarItems <- colnames(itemcosine)[
    order(itemcosine[, usr_stockcode], decreasing = TRUE)[2:6]]

# Get descriptions
Top5SimilarItemsDescriptions <- unique(
    dat_obj[which(dat_obj$StockNo %in% Top5SimilarItems), c("StockNo", "Item.Descr
iption")])

Top5SimilarItemsDescriptions <- Top5SimilarItemsDescriptions[
    match(Top5SimilarItems, Top5SimilarItemsDescriptions$StockNo),]

Top5SimilarItemsDescriptions

##         StockNo                Item.Description
## 6097 1021ST2082           EMB. GT-85540 GEO 555
## 6099  1021ST425 DYED EMB. RC-1101611 GEO AAA MP
## 5384  1121ST388         DIG GB1772 COTTON SUPER
## 3148  1121ST394         DIG GB1790 COTTON SUPER
## 3143  1121ST405         DIG GB1791 COTTON SUPER
```

Conclusion

Exploratory Data Analysis:

The top-selling products (Top sellers) have been compiled and visualized in a histogram displaying the quantity versus each product.

Our analysis (Order-wise analysis) of orders and revenue trends reveals a prominent seasonal pattern.

As per Repeat Customers analysis, noted that most customers are one-time purchasers, with a smaller fraction being repeat customers. Therefore, it is imperative to devise a strategy aimed at increasing customer retention, considering the significance of the 80-20 rule as previously elucidated.

Cosine similarity effectively enables the recommendation of additional items based on user similarity, enhancing personalized recommendations.

Cosine similarity facilitates accurate item recommendations by leveraging item similarity, enhancing the relevance and diversity of suggested items.

● → Click to follow link

Citation

1.Data-set
2.Cosin-Similarity Math
3.Movie recommendation system