1. Which dataset you’re using
2. Specification of your dataset
3. Pre-processing
4. How K-means / other clustering algorithm work
5. Result of implementation
6. Conclusion

**Online Retail II Data Set**

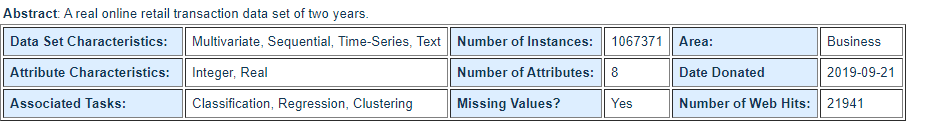
**Online Retail**

**DATA SET DESCRIPTION**

This is a transnational data set which contains all the transactions occurring between 12/01/2010 and 12/09/2011 for a UK-based and registered non-store online retail.The company mainly sells unique all-occasion gifts. Many customers of the company are wholesalers.

<https://archive.ics.uci.edu/ml/datasets/Online+Retail+II>

2. Specification of Online Retail II Data set



**Data Set Information:**

This Online Retail II data set contains all the transactions occurring for a UK-based and registered, non-store online retail between 01/12/2009 and 09/12/2011.The company mainly sells unique all-occasion gift-ware. Many customers of the company are wholesalers.

n relation to customer-centric business intelligence, online retailers are usually concerned with the following common business concerns:

* Which items/products’ web pages has a customer visited? How long has a customer stayed with each web page, and in which sequence has a customer visited a set of products’ web pages?
* Who are the most/least valuable customers to the business? What are the distinct characteristics of them?
* Who are the most/least loyal customers, and how are they characterized?
* What are customers’ purchase behaviour patterns? Which products/items have customers purchased together often? In which sequence the products have been purchased?
* Which types of customers are more likely to respond to a certain promotion mailing? and
* What are the sales patterns in terms of various perspectives such as products/items, regions and time (weekly, monthly, quarterly, yearly and seasonally), and so on?

In order to address these business concerns, data mining techniques have been widely adopted across the online retail sector, coupled with a set of well-known business metrics about customers’ profitability and values, for instance, the recency, frequency and monetary (RFM) model,[2](https://link.springer.com/article/10.1057/dbm.2012.17#CR2)and the customer life value model.[3](https://link.springer.com/article/10.1057/dbm.2012.17#CR3)For many online retailers in the United Kingdom and internationally alike, especially the leading companies including Amazon, Walmart, Tesco, Sainsbury's, Argos, Marks and Spencer, John Lewis, and EasyJet, data mining has now become a common practice and an integral part of the business processes in creating customer-centric business intelligence and supporting customer-centric marketing.[4](https://link.springer.com/article/10.1057/dbm.2012.17#CR4),[5](https://link.springer.com/article/10.1057/dbm.2012.17#CR5)

Although many famous online retail brands are embracing data mining techniques as crucial tools to gain competitive advantages on the market, there are still many smaller ones and new entrants are keen to practise consumer-centric marketing yet technically lack the necessary knowledge and expertise to do so.

In this article a case study of using data mining techniques in customer-centric business intelligence for an online retailer is presented. The online retailer considered here is a typical one: a small business and a relatively new entrant to the online retail sector, knowing the growing importance of being analytical in today's online businesses and data mining techniques, however, lacking technical awareness and recourses. The main purpose of this analysis is to help the business better understand its customers and therefore conduct customer-centric marketing more effectively. On the basis of the RFM model, customers of the business have been segmented into various meaningful groups using the *k*-means clustering algorithm and decision tree induction, and the main characteristics of the consumers in each segment have been clearly identified. Accordingly, a set of recommendations is provided to the business on customer-centric marketing and further data analysis tasks. The analysis is developed in a step-by-step way. SAS Enterprise Guide and SAS Enterprise Miner[6](https://link.springer.com/article/10.1057/dbm.2012.17#CR6),[7](https://link.springer.com/article/10.1057/dbm.2012.17#CR7),[8](https://link.springer.com/article/10.1057/dbm.2012.17#CR8),[9](https://link.springer.com/article/10.1057/dbm.2012.17#CR9) have been employed in this study.

The rest of this article is organized as follows. The next section provides the background information about the online retailer studied in the article along with the associated dataset to be explored. The section after that discusses in detail about the main steps and tasks for data pre-processing in order to create an appropriate target dataset for the required further analyses. In the subsequent section the *k*-means clustering analysis is performed and a set of meaningful clusters and segments of the target dataset has been identified. A detailed discussion on each of the clusters is given, and the segmentation is further refined by using decision tree induction. The penultimate section summarizes the essential consumer-centric business intelligence based on the analysis results, and provides some concrete recommendations to the online retailer aiming at maximizing profits for the business. Finally the concluding remarks are given in the last section.

Url= <https://archive.ics.uci.edu/ml/machine-learning-databases/00502/>[online\_retail\_II.xlsx](https://archive.ics.uci.edu/ml/machine-learning-databases/00502/online_retail_II.xlsx)

Df<-read/

**Attribute Information:**

InvoiceNo: Invoice number. Nominal. A 6-digit integral number uniquely assigned to each transaction. If this code starts with the letter 'c', it indicates a cancellation.  
StockCode: Product (item) code. Nominal. A 5-digit integral number uniquely assigned to each distinct product.  
Description: Product (item) name. Nominal.  
Quantity: The quantities of each product (item) per transaction. Numeric.  
InvoiceDate: Invice date and time. Numeric. The day and time when a transaction was generated.  
UnitPrice: Unit price. Numeric. Product price per unit in sterling (Â£).  
CustomerID: Customer number. Nominal. A 5-digit integral number uniquely assigned to each customer.  
Country: Country name. Nominal. The name of the country where a customer resides.

## Load the data

The raw data includes 541909 observations of 8 variables. I omitted 136534 missing observations and the new data set has 406829 observations

## Data pre-processing

Create an aggregated variable named Amount, by multiplying Quantity with Price, which gives the total amount of money spent per product / item in each transaction. The amount variable is assume to represent sales revenue. Separate the variable InvoiceDate into two variables Date and Time . This allows different transactions created by the same consumer on the same day but at different times to be treated separately. Create SKU variable that encodes the first 3 digits of Stockcode to indicate different stock keeping unit, which is related to later calculation of "Breadth" customer behavior indicator.

#omit NA

eRetail<-na.omit(eRetail)

#create Amount variable#

eRetail$Amount <- eRetail$Quantity \* eRetail$UnitPrice

#create SKU variable

eRetail$SKU <- substr(eRetail$StockCode,1,3)

#separate date & time#

eRetail$InvoiceDate<-strptime(eRetail$InvoiceDate,"%m/%d/%Y %H:%M")

eRetail$InvoiceTime = format(eRetail$InvoiceDate,"%H")

eRetail$InvoiceDate<-as.Date(eRetail$InvoiceDate,"%m/%d/%Y")

#look at internal structure#

str(eRetail)

## 'data.frame': 406829 obs. of 11 variables:

## $ InvoiceNo : Factor w/ 25900 levels "536365","536366",..: 1 1 1 1 1 1 1 2 2 3 ...

## $ StockCode : Factor w/ 4070 levels "10002","10080",..: 3538 2795 3045 2986 2985 1663 801 1548 1547 3306 ...

## $ Description: Factor w/ 4223 levels " 4 PURPLE FLOCK DINNER CANDLES",..: 4026 4034 931 1958 2979 3234 1572 1697 1694 258 ...

## $ Quantity : int 6 6 8 6 6 2 6 6 6 32 ...

## $ InvoiceDate: Date, format: "2010-12-01" "2010-12-01" ...

## $ UnitPrice : num 2.55 3.39 2.75 3.39 3.39 7.65 4.25 1.85 1.85 1.69 ...

## $ CustomerID : int 17850 17850 17850 17850 17850 17850 17850 17850 17850 13047 ...

## $ Country : Factor w/ 38 levels "Australia","Austria",..: 36 36 36 36 36 36 36 36 36 36 ...

## $ Amount : num 15.3 20.3 22 20.3 20.3 ...

## $ SKU : chr "851" "710" "844" "840" ...

## $ InvoiceTime: chr "08" "08" "08" "08" ...

## - attr(\*, "na.action")=Class 'omit' Named int [1:135080] 623 1444 1445 1446 1447 1448 1449 1450 1451 1452 ...

## .. ..- attr(\*, "names")= chr [1:135080] "623" "1444" "1445" "1446" ...

#View a summary#

summary(eRetail)

## InvoiceNo StockCode

## 576339 : 542 85123A : 2077

## 579196 : 533 22423 : 1905

## 580727 : 529 85099B : 1662

## 578270 : 442 84879 : 1418

## 573576 : 435 47566 : 1416

## 567656 : 421 20725 : 1359

## (Other):403927 (Other):396992

## Description Quantity

## WHITE HANGING HEART T-LIGHT HOLDER: 2070 Min. :-80995.00

## REGENCY CAKESTAND 3 TIER : 1905 1st Qu.: 2.00

## JUMBO BAG RED RETROSPOT : 1662 Median : 5.00

## ASSORTED COLOUR BIRD ORNAMENT : 1418 Mean : 12.06

## PARTY BUNTING : 1416 3rd Qu.: 12.00

## LUNCH BAG RED RETROSPOT : 1358 Max. : 80995.00

## (Other) :397000

## InvoiceDate UnitPrice CustomerID

## Min. :2010-12-01 Min. : 0.00 Min. :12346

## 1st Qu.:2011-04-06 1st Qu.: 1.25 1st Qu.:13953

## Median :2011-07-31 Median : 1.95 Median :15152

## Mean :2011-07-10 Mean : 3.46 Mean :15288

## 3rd Qu.:2011-10-20 3rd Qu.: 3.75 3rd Qu.:16791

## Max. :2011-12-09 Max. :38970.00 Max. :18287

##

## Country Amount SKU

## United Kingdom:361878 Min. :-168469.6 Length:406829

## Germany : 9495 1st Qu.: 4.2 Class :character

## France : 8491 Median : 11.1 Mode :character

## EIRE : 7485 Mean : 20.4

## Spain : 2533 3rd Qu.: 19.5

## Netherlands : 2371 Max. : 168469.6

## (Other) : 14576

## InvoiceTime

## Length:406829

## Class :character

## Mode :character

##

##

##

##

#View the top#

head(eRetail)

## InvoiceNo StockCode Description Quantity

## 1 536365 85123A WHITE HANGING HEART T-LIGHT HOLDER 6

## 2 536365 71053 WHITE METAL LANTERN 6

## 3 536365 84406B CREAM CUPID HEARTS COAT HANGER 8

## 4 536365 84029G KNITTED UNION FLAG HOT WATER BOTTLE 6

## 5 536365 84029E RED WOOLLY HOTTIE WHITE HEART. 6

## 6 536365 22752 SET 7 BABUSHKA NESTING BOXES 2

## InvoiceDate UnitPrice CustomerID Country Amount SKU InvoiceTime

## 1 2010-12-01 2.55 17850 United Kingdom 15.30 851 08

## 2 2010-12-01 3.39 17850 United Kingdom 20.34 710 08

## 3 2010-12-01 2.75 17850 United Kingdom 22.00 844 08

## 4 2010-12-01 3.39 17850 United Kingdom 20.34 840 08

## 5 2010-12-01 3.39 17850 United Kingdom 20.34 840 08

## 6 2010-12-01 7.65 17850 United Kingdom 15.30 227 08

# SIMPLE EXPLORATION OF DATA

# Outliers

## boxplot of Amount

boxplot(eRetail$Amount)$stats[c(1, 5), ]

## [1] -18.75 42.45

#cutoff outliner

Retail<-subset(eRetail,eRetail$Amount>= 0 & eRetail$Amount<= 10000 )

boxplot(Retail$Amount)$stats[c(1, 5), ]

## [1] 0.00 42.45

## What are top 5 selling products accross all times?

Retail1 <- ddply(Retail, .(StockCode,Description), summarize, sumAmount= sum(Amount), sumQuantity= sum(Quantity), nCustomer= length(unique(CustomerID)), nPurchase= length(unique(InvoiceNo)) )

head(Retail1[order(-Retail1$sumQuantity),] )

## StockCode Description sumAmount sumQuantity

## 3020 84077 WORLD WAR 2 GLIDERS ASSTD DESIGNS 13586.25 54415

## 3444 85099B JUMBO BAG RED RETROSPOT 85220.78 46181

## 3459 85123A WHITE HANGING HEART T-LIGHT HOLDER 100448.15 36725

## 3278 84879 ASSORTED COLOUR BIRD ORNAMENT 56580.34 35362

## 433 21212 PACK OF 72 RETROSPOT CAKE CASES 16394.53 33693

## 1109 22197 POPCORN HOLDER 23427.71 30931

## nCustomer nPurchase

## 3020 307 472

## 3444 635 1600

## 3459 856 1971

## 3278 678 1375

## 433 635 1029

## 1109 295 632

head(Retail1[order(-Retail1$nCustomer),] )

## StockCode Description sumAmount sumQuantity

## 1319 22423 REGENCY CAKESTAND 3 TIER 142592.95 12412

## 3459 85123A WHITE HANGING HEART T-LIGHT HOLDER 100448.15 36725

## 2799 47566 PARTY BUNTING 68844.33 15295

## 3278 84879 ASSORTED COLOUR BIRD ORNAMENT 56580.34 35362

## 1608 22720 SET OF 3 CAKE TINS PANTRY DESIGN 33347.80 7020

## 433 21212 PACK OF 72 RETROSPOT CAKE CASES 16394.53 33693

## nCustomer nPurchase

## 1319 881 1704

## 3459 856 1971

## 2799 708 1380

## 3278 678 1375

## 1608 640 1146

## 433 635 1029

head(Retail1[order(-Retail1$sumAmount),] )

## StockCode Description sumAmount sumQuantity

## 1319 22423 REGENCY CAKESTAND 3 TIER 142592.95 12412

## 3459 85123A WHITE HANGING HEART T-LIGHT HOLDER 100448.15 36725

## 3444 85099B JUMBO BAG RED RETROSPOT 85220.78 46181

## 3896 POST POSTAGE 77803.96 3120

## 2799 47566 PARTY BUNTING 68844.33 15295

## 3278 84879 ASSORTED COLOUR BIRD ORNAMENT 56580.34 35362

## nCustomer nPurchase

## 1319 881 1704

## 3459 856 1971

## 3444 635 1600

## 3896 331 1099

## 2799 708 1380

## 3278 678 1375

head(Retail1[order(-Retail1$nPurchase),] )

## StockCode Description sumAmount sumQuantity

## 3459 85123A WHITE HANGING HEART T-LIGHT HOLDER 100448.15 36725

## 1319 22423 REGENCY CAKESTAND 3 TIER 142592.95 12412

## 3444 85099B JUMBO BAG RED RETROSPOT 85220.78 46181

## 2799 47566 PARTY BUNTING 68844.33 15295

## 3278 84879 ASSORTED COLOUR BIRD ORNAMENT 56580.34 35362

## 175 20725 LUNCH BAG RED RETROSPOT 28048.45 17697

## nCustomer nPurchase

## 3459 856 1971

## 1319 881 1704

## 3444 635 1600

## 2799 708 1380

## 3278 678 1375

## 175 532 1288

## Do these sales of top selling products change with time (months)?Any seasonality?

Retail2 <- subset(Retail, Description%in%c("MEDIUM CERAMIC TOP STORAGE JAR","JUMBO BAG RED RETROSPOT","REGENCY CAKESTAND 3 TIER","WHITE HANGING HEART T-LIGHT HOLDER","PARTY BUNTING","WORLD WAR 2 GLIDERS ASSTD DESIGNS"), select = c(Description,InvoiceDate,InvoiceTime,Quantity,CustomerID,Amount,InvoiceNo))

Retail2$Invoice\_month<-month(Retail2$InvoiceDate)

Retail2$Decription<-as.character(Retail2$Description)

ggplot(Retail2, aes(x=Invoice\_month, y= Quantity))+ facet\_wrap(~Description, ncol=2) +

geom\_bar(stat="identity") +

labs(title = "Sales by month", x = "Month", y = "Sales Volume")

ggplot(Retail2, aes(x=Invoice\_month, y= length(unique(CustomerID)) )) + facet\_wrap(~Description, ncol=2) +

geom\_bar(stat="identity") +

labs(title = "Sales by month", x = "Month", y = "Number of Customer")

ggplot(Retail2, aes(x=Invoice\_month, y= Amount )) + facet\_wrap(~Description, ncol=2) +

geom\_bar(stat="identity") +

labs(title = "Sales by month", x = "Month", y = "Sales Revenue")

ggplot(Retail2, aes(x=Invoice\_month, y= length(unique(InvoiceNo)) )) + facet\_wrap(~Description, ncol=2) +

geom\_bar(stat="identity") +

labs(title = "Sales by month", x = "Month", y = "Number of Purchases")

\*The sales of products changes with time.

## What are the busiest hours of a day?

Retail3<-ddply(Retail, .(InvoiceTime), summarize, sumAmount=sum(Amount), sumQuantity=sum(Quantity), nCustomer=length(unique(CustomerID)))

names(Retail3) [1] <-"InvoiceHour"

ggplot(Retail3, aes(x=InvoiceHour, y= sumQuantity)) +

geom\_bar(stat="identity") +

labs(title = "Sales by hours", x = "Hours", y = "Sales Volume")

ggplot(Retail3, aes(x=InvoiceHour, y= nCustomer)) +

geom\_bar(stat="identity") +

labs(title = "Sales by hours", x = "Hours", y = "Number of customer")

 \*The busiest hour of the day is around 12 pm for sales volume, and 12 pm for number of customers.

# START RFM ANALYSIS

## Building dataset for RFM analysis

getRFMdf<-function (RFM\_raw){

RFM\_raw <- RFM\_raw[!duplicated(RFM\_raw$CustomerID),]

RFM\_raw <- cbind(RFM\_raw, First\_date = with(df,

as.Date(as.integer(by(InvoiceDate, CustomerID, min)), "1970/01/01")))

RFM\_raw <- cbind(RFM\_raw, Last\_date = with(df,

as.Date(as.integer(by(InvoiceDate, CustomerID, max)), "1970/01/01")))

#Recency

AsOfDate <- max(RFM\_raw$Last\_date)

RFM\_raw <- cbind(RFM\_raw, Recency = with(df,

as.numeric(difftime(AsOfDate,RFM\_raw$Last\_date,units="days")))/30)

#First\_purchase

RFM\_raw <- cbind(RFM\_raw, First\_purchase = with(df,

as.numeric(difftime(AsOfDate,RFM\_raw$First\_date,units="days")))/30)

#Frequency

RFM\_raw <- cbind(RFM\_raw, Frequency = with(df,

as.numeric(by(InvoiceNo, CustomerID, function(x) length(unique(x))))))

#Monetary & related

RFM\_raw <- cbind(RFM\_raw, Monetary = with(df,

as.numeric(by(Amount, CustomerID, sum))))

RFM\_raw <- cbind(RFM\_raw, AvgM = with(df,

as.numeric(by(Amount, CustomerID, mean))))

RFM\_raw <- cbind(RFM\_raw, maxM = with(df,

as.numeric(by(Amount, CustomerID, max))))

#Breadth

RFM\_raw <- cbind(RFM\_raw, Breadth = with(df,

as.numeric(by(SKU, CustomerID, function(x) length(unique(x))))))

#Tenure

RFM\_raw <- cbind(RFM\_raw, Tenure = with(df, as.numeric(difftime(RFM\_raw$Last\_date,RFM\_raw$First\_date,units="days")))/30)

#sum Quantity

RFM\_raw <- cbind(RFM\_raw, sumQuant = with(df,

as.numeric(by(Quantity, CustomerID, mean))))

}

## define getRFMnor

getRFMnor<-function (RFMn){

RFMn<- as.data.frame(scale(df2[14:22], center= TRUE))

RFMn<- cbind(df2[,c(1:13)],RFMn)

RFMn<- rename(RFMn, c("Recency" = "R", "Frequency" = "Fq", "Monetary" = "M", "Breadth" = "B" , "Tenure" = "Ten", "sumQuant" = "Q" ) )

}

## define getRFMscore function

#score 1 to 5

score15<-function(x){

ceiling((rank(x))/(length(x))\*5)

}

getRFMscore<-function (RFMs){

RFMs <- as.data.frame(lapply(df3[,c(15:22)], score15))

RFMs <- cbind(df3[,c(1:13)], R= ceiling((rank(-df3$R))/(length(df3$R))\*5), RFMs)

RFMs <- cbind(RFMs,RFMScore = 100\*RFMs$R + 10\*RFMs$Fq+RFMs$M)

}

#### RFM score on the whole data (12/2010 to 12/2011)

df<- eRetail

rawRFM<-as.data.frame(getRFMdf(df))

#### some rawRFM score EDA

#take a look at disturbution

par(mfrow = c(1,3))

boxplot(rawRFM$Recency)$stats[c(1, 5), ]

## [1] 0.0 11.1

boxplot(rawRFM$Frequency)$stats[c(1, 5), ]

## [1] 1 11

boxplot(rawRFM$Monetary)$stats[c(1, 5), ]

## [1] -1592.49 3580.13

Strong left-skewness for Recency, Frequency, TotalAmount,Monetary,Breadth, and Tenure.

#### Exclude outliners

RFM<-subset(rawRFM,rawRFM$Recency<= 12 & rawRFM$Frequency<= 25 & rawRFM$Monetary>= 0 & rawRFM$Monetary<= 10000)

summary(rawRFM$Monetary)

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

## -4288.0 293.4 648.1 1898.0 1612.0 279500.0

par(mfrow = c(1,3))

hist(RFM$Recency)

hist(RFM$Frequency)

hist(RFM$Monetary)

Now, the Left-skewness is better.

#take a look at disturbution

par(mfrow = c(1,3))

boxplot(RFM$Recency)$stats[c(1, 5), ]

## [1] 0.0 10.3

boxplot(RFM$Frequency)$stats[c(1, 5), ]

## [1] 1 11

boxplot(RFM$Monetary)$stats[c(1, 5), ]

## [1] 0.00 3250.64

#data normalization

df2<- RFM

nRFM<-as.data.frame(getRFMnor(df2))

#score

df3 <- nRFM

RFMs<-as.data.frame(getRFMscore(df3))

par(mfrow = c(1,3))

hist(RFMs$R)

hist(RFMs$Fq)

hist(RFMs$M)

# K-means cluster analysis

## Visulize the clusters

RFM\_cluster <- data.frame(nRFM$R,nRFM$Fq,nRFM$M)

km <- kmeans(RFM\_cluster,centers=5)

RFM\_cluster$cluster <- as.factor(km$cluster)

RFM\_cluster <- cbind(RFM\_cluster,RFMs)

ggplot(RFM\_cluster,aes(x=nRFM.R, y=nRFM.M, color= cluster,size= nRFM.Fq))+geom\_point()+ scale\_size\_area(max\_size=10)+labs(x="Recency", y="Monetary")

## Warning: Removed 2877 rows containing missing values (geom\_point).

ggplot(RFM\_cluster,aes(x=R, y= M, color= cluster,size= Fq))+geom\_point()+ scale\_size\_area(max\_size=20)+labs(x="Recency", y="Monetary")

# separate clusters

RFM\_cluster1<-RFM\_cluster[which(RFM\_cluster$cluster==1),]

ggplot(RFM\_cluster1,aes(x=R, y=M, color= Fq, size = 10))+geom\_point()+ labs(x="Recency", y="Monetary")

apply(RFM\_cluster1[,c(18,20:27)],2,mean)

## R Fq M AvgM maxM B

## 1.000000 1.457961 1.826476 3.048301 2.536673 1.955277

## Ten Q RFMScore

## 1.443649 2.783542 116.406082

ggplot(RFM\_cluster1,aes(x=RFMScore))+geom\_histogram(bins=50)+ labs(x="RFMScore", y="Count")

RFM\_cluster2<-RFM\_cluster[which(RFM\_cluster$cluster==2),]

ggplot(RFM\_cluster2,aes(x=R, y=M, color= Fq, size = 10))+geom\_point()+ labs(x="Recency", y="Monetary")

apply(RFM\_cluster2[,c(18,20:27)],2,mean)

## R Fq M AvgM maxM B

## 3.891827 4.537260 4.591346 3.165865 3.765625 4.165865

## Ten Q RFMScore

## 4.271635 3.216346 439.146635

ggplot(RFM\_cluster2,aes(x=RFMScore))+geom\_histogram(bins=50)+ labs(x="RFMScore", y="Count")

RFM\_cluster3<-RFM\_cluster[which(RFM\_cluster$cluster==3),]

ggplot(RFM\_cluster3,aes(x=R, y=M, color= Fq, size = 10))+geom\_point()+ labs(x="Recency", y="Monetary")

apply(RFM\_cluster3[,c(18,20:27)],2,mean)

## R Fq M AvgM maxM B

## 4.365759 4.933852 4.996109 3.498054 4.291829 4.389105

## Ten Q RFMScore

## 4.715953 3.463035 490.910506

ggplot(RFM\_cluster3,aes(x=RFMScore))+geom\_histogram(bins=50)+ labs(x="RFMScore", y="Count")

RFM\_cluster4<-RFM\_cluster[which(RFM\_cluster$cluster==4),]

ggplot(RFM\_cluster4,aes(x=R, y=M, color= Fq, size = 10))+geom\_point()+ labs(x="Recency", y="Monetary")

apply(RFM\_cluster4[,c(18,20:27)],2,mean)

## R Fq M AvgM maxM B

## 1.620329 2.143498 2.379671 3.073244 2.799701 2.430493

## Ten Q RFMScore

## 2.239163 2.979073 185.847534

ggplot(RFM\_cluster4,aes(x=RFMScore))+geom\_histogram(bins=50)+ labs(x="RFMScore", y="Count")

RFM\_cluster5<-RFM\_cluster[which(RFM\_cluster$cluster==5),]

ggplot(RFM\_cluster5,aes(x=R, y=M, color= Fq, size = 10))+geom\_point()+ labs(x="Recency", y="Monetary")

apply(RFM\_cluster5[,c(18,20:27)],2,mean)

## R Fq M AvgM maxM B

## 3.525641 2.326644 2.574136 2.810479 2.698997 2.840022

## Ten Q RFMScore

## 2.680045 2.910256 378.404682

ggplot(RFM\_cluster5,aes(x=RFMScore))+geom\_histogram(bins=50)+ labs(x="RFMScore", y="Count")

## For each cluster, What are top 5 selling products accross all times?

cluster1 <- ddply(RFM\_cluster1, .(StockCode,Description), summarize, sumAmount= sum(Amount), sumQuantity= sum(Quantity), nCustomer= length(unique(CustomerID)), nPurchase= length(unique(InvoiceNo)) )

head(cluster1[order(-cluster1$sumQuantity),] )

## StockCode Description sumAmount sumQuantity

## 394 84077 WORLD WAR 2 GLIDERS ASSTD DESIGNS 604.80 2880

## 385 79321 CHILLI LIGHTS 2068.80 512

## 222 22616 PACK OF 12 LONDON TISSUES 95.04 432

## 375 62018 SOMBRERO 500.00 400

## 125 22084 PAPER CHAIN KIT EMPIRE 912.30 354

## 383 75049L LARGE CIRCULAR MIRROR MOBILE 259.80 300

## nCustomer nPurchase

## 394 2 2

## 385 3 3

## 222 1 1

## 375 1 1

## 125 2 2

## 383 2 2

head(cluster1[order(-cluster1$nCustomer),] )

## StockCode Description sumAmount sumQuantity

## 426 85123A WHITE HANGING HEART T-LIGHT HOLDER 732.05 275

## 179 22423 REGENCY CAKESTAND 3 TIER 216.75 17

## 148 22178 VICTORIAN GLASS HANGING T-LIGHT 60.45 45

## 34 21034 REX CASH+CARRY JUMBO SHOPPER 10.45 11

## 126 22086 PAPER CHAIN KIT 50'S CHRISTMAS 279.00 100

## 135 22139 RETROSPOT TEA SET CERAMIC 11 PC 172.00 38

## nCustomer nPurchase

## 426 10 10

## 179 7 7

## 148 5 5

## 34 4 4

## 126 4 4

## 135 4 4

head(cluster1[order(-cluster1$sumAmount),] )

## StockCode Description sumAmount sumQuantity

## 385 79321 CHILLI LIGHTS 2068.80 512

## 125 22084 PAPER CHAIN KIT EMPIRE 912.30 354

## 426 85123A WHITE HANGING HEART T-LIGHT HOLDER 732.05 275

## 394 84077 WORLD WAR 2 GLIDERS ASSTD DESIGNS 604.80 2880

## 267 22826 LOVE SEAT ANTIQUE WHITE METAL 535.00 5

## 311 23084 RABBIT NIGHT LIGHT 523.84 292

## nCustomer nPurchase

## 385 3 3

## 125 2 2

## 426 10 10

## 394 2 2

## 267 3 3

## 311 2 2

head(cluster1[order(-cluster1$nPurchase),] )

## StockCode Description sumAmount sumQuantity

## 426 85123A WHITE HANGING HEART T-LIGHT HOLDER 732.05 275

## 179 22423 REGENCY CAKESTAND 3 TIER 216.75 17

## 148 22178 VICTORIAN GLASS HANGING T-LIGHT 60.45 45

## 34 21034 REX CASH+CARRY JUMBO SHOPPER 10.45 11

## 126 22086 PAPER CHAIN KIT 50'S CHRISTMAS 279.00 100

## 135 22139 RETROSPOT TEA SET CERAMIC 11 PC 172.00 38

## nCustomer nPurchase

## 426 10 10

## 179 7 7

## 148 5 5

## 34 4 4

## 126 4 4

## 135 4 4

cluster2 <- ddply(RFM\_cluster2, .(StockCode,Description), summarize, sumAmount= sum(Amount), sumQuantity= sum(Quantity), nCustomer= length(unique(CustomerID)), nPurchase= length(unique(InvoiceNo)) )

head(cluster2[order(-cluster2$sumQuantity),] )

## StockCode Description sumAmount sumQuantity

## 9 16014 SMALL CHINESE STYLE SCISSOR 320.00 1000

## 171 22086 PAPER CHAIN KIT 50'S CHRISTMAS 892.70 346

## 566 85123A WHITE HANGING HEART T-LIGHT HOLDER 808.55 313

## 571 85152 HAND OVER THE CHOCOLATE SIGN 558.00 300

## 85 21422 PORCELAIN ROSE SMALL 207.36 288

## 547 84949 SILVER HANGING T-LIGHT HOLDER 417.60 288

## nCustomer nPurchase

## 9 1 1

## 171 4 4

## 566 7 7

## 571 2 2

## 85 1 1

## 547 1 1

head(cluster2[order(-cluster2$nCustomer),] )

## StockCode Description sumAmount sumQuantity

## 241 22423 REGENCY CAKESTAND 3 TIER 2320.80 208

## 7 15056BL EDWARDIAN PARASOL BLACK 630.20 116

## 566 85123A WHITE HANGING HEART T-LIGHT HOLDER 808.55 313

## 402 22961 JAM MAKING SET PRINTED 104.40 72

## 523 82494L WOODEN FRAME ANTIQUE WHITE 333.30 126

## 541 84879 ASSORTED COLOUR BIRD ORNAMENT 367.20 240

## nCustomer nPurchase

## 241 16 16

## 7 7 7

## 566 7 7

## 402 5 5

## 523 5 5

## 541 5 5

head(cluster2[order(-cluster2$sumAmount),] )

## StockCode Description sumAmount sumQuantity

## 241 22423 REGENCY CAKESTAND 3 TIER 2320.80 208

## 171 22086 PAPER CHAIN KIT 50'S CHRISTMAS 892.70 346

## 566 85123A WHITE HANGING HEART T-LIGHT HOLDER 808.55 313

## 7 15056BL EDWARDIAN PARASOL BLACK 630.20 116

## 511 51008 AFGHAN SLIPPER SOCK PAIR 590.00 200

## 336 22777 GLASS CLOCHE LARGE 567.80 74

## nCustomer nPurchase

## 241 16 16

## 171 4 4

## 566 7 7

## 7 7 7

## 511 1 1

## 336 2 2

head(cluster2[order(-cluster2$nPurchase),] )

## StockCode Description sumAmount sumQuantity

## 241 22423 REGENCY CAKESTAND 3 TIER 2320.80 208

## 7 15056BL EDWARDIAN PARASOL BLACK 630.20 116

## 566 85123A WHITE HANGING HEART T-LIGHT HOLDER 808.55 313

## 402 22961 JAM MAKING SET PRINTED 104.40 72

## 523 82494L WOODEN FRAME ANTIQUE WHITE 333.30 126

## 541 84879 ASSORTED COLOUR BIRD ORNAMENT 367.20 240

## nCustomer nPurchase

## 241 16 16

## 7 7 7

## 566 7 7

## 402 5 5

## 523 5 5

## 541 5 5

cluster3 <- ddply(RFM\_cluster3, .(StockCode,Description), summarize, sumAmount= sum(Amount), sumQuantity= sum(Quantity), nCustomer= length(unique(CustomerID)), nPurchase= length(unique(InvoiceNo)) )

head(cluster3[order(-cluster3$sumQuantity),] )

## StockCode Description sumAmount sumQuantity

## 193 84077 WORLD WAR 2 GLIDERS ASSTD DESIGNS 532.32 2928

## 107 22616 PACK OF 12 LONDON TISSUES 324.00 1296

## 202 84945 MULTI COLOUR SILVER T-LIGHT HOLDER 276.48 384

## 212 85099C JUMBO BAG BAROQUE BLACK WHITE 358.00 200

## 159 23211 RED ROCKING HORSE HAND PAINTED 149.76 144

## 186 79321 CHILLI LIGHTS 408.00 96

## nCustomer nPurchase

## 193 2 2

## 107 1 1

## 202 1 1

## 212 1 1

## 159 1 1

## 186 2 2

head(cluster3[order(-cluster3$nCustomer),] )

## StockCode Description sumAmount sumQuantity

## 85 22423 REGENCY CAKESTAND 3 TIER 424.55 49

## 214 85123A WHITE HANGING HEART T-LIGHT HOLDER 199.60 72

## 6 20724 RED RETROSPOT CHARLOTTE BAG 25.50 30

## 60 22086 PAPER CHAIN KIT 50'S CHRISTMAS 159.30 54

## 90 22469 HEART OF WICKER SMALL 52.80 32

## 200 84879 ASSORTED COLOUR BIRD ORNAMENT 81.12 48

## nCustomer nPurchase

## 85 8 8

## 214 6 6

## 6 3 3

## 60 3 3

## 90 3 3

## 200 3 3

head(cluster3[order(-cluster3$sumAmount),] )

## StockCode Description sumAmount sumQuantity

## 193 84077 WORLD WAR 2 GLIDERS ASSTD DESIGNS 532.32 2928

## 85 22423 REGENCY CAKESTAND 3 TIER 424.55 49

## 186 79321 CHILLI LIGHTS 408.00 96

## 212 85099C JUMBO BAG BAROQUE BLACK WHITE 358.00 200

## 107 22616 PACK OF 12 LONDON TISSUES 324.00 1296

## 202 84945 MULTI COLOUR SILVER T-LIGHT HOLDER 276.48 384

## nCustomer nPurchase

## 193 2 2

## 85 8 8

## 186 2 2

## 212 1 1

## 107 1 1

## 202 1 1

head(cluster3[order(-cluster3$nPurchase),] )

## StockCode Description sumAmount sumQuantity

## 85 22423 REGENCY CAKESTAND 3 TIER 424.55 49

## 214 85123A WHITE HANGING HEART T-LIGHT HOLDER 199.60 72

## 6 20724 RED RETROSPOT CHARLOTTE BAG 25.50 30

## 60 22086 PAPER CHAIN KIT 50'S CHRISTMAS 159.30 54

## 90 22469 HEART OF WICKER SMALL 52.80 32

## 200 84879 ASSORTED COLOUR BIRD ORNAMENT 81.12 48

## nCustomer nPurchase

## 85 8 8

## 214 6 6

## 6 3 3

## 60 3 3

## 90 3 3

## 200 3 3

cluster4 <- ddply(RFM\_cluster4, .(StockCode,Description), summarize, sumAmount= sum(Amount), sumQuantity= sum(Quantity), nCustomer= length(unique(CustomerID)), nPurchase= length(unique(InvoiceNo)) )

head(cluster4[order(-cluster4$sumQuantity),] )

## StockCode Description sumAmount sumQuantity

## 11 17096 ASSORTED LAQUERED INCENSE HOLDERS 293.76 1728

## 343 23167 SMALL CERAMIC TOP STORAGE JAR 931.50 1350

## 9 17003 BROCADE RING PURSE 180.00 720

## 454 84947 ANTIQUE SILVER TEA GLASS ENGRAVED 389.10 366

## 227 22560 TRADITIONAL MODELLING CLAY 265.00 250

## 414 71459 HANGING JAM JAR T-LIGHT HOLDER 179.89 241

## nCustomer nPurchase

## 11 1 1

## 343 1 1

## 9 1 1

## 454 2 2

## 227 1 1

## 414 4 4

head(cluster4[order(-cluster4$nCustomer),] )

## StockCode Description sumAmount sumQuantity

## 201 22423 REGENCY CAKESTAND 3 TIER 941.40 84

## 16 20685 DOORMAT RED RETROSPOT 96.30 12

## 485 M Manual -5719.22 0

## 135 22086 PAPER CHAIN KIT 50'S CHRISTMAS 307.25 115

## 215 22501 PICNIC BASKET WICKER LARGE 169.15 17

## 403 47566 PARTY BUNTING 151.95 31

## nCustomer nPurchase

## 201 7 7

## 16 6 6

## 485 6 6

## 135 5 5

## 215 5 5

## 403 5 5

head(cluster4[order(-cluster4$sumAmount),] )

## StockCode Description sumAmount sumQuantity

## 201 22423 REGENCY CAKESTAND 3 TIER 941.40 84

## 343 23167 SMALL CERAMIC TOP STORAGE JAR 931.50 1350

## 153 22171 3 HOOK PHOTO SHELF ANTIQUE WHITE 734.40 96

## 431 84078A SET/4 WHITE RETRO STORAGE CUBES 699.00 20

## 454 84947 ANTIQUE SILVER TEA GLASS ENGRAVED 389.10 366

## 135 22086 PAPER CHAIN KIT 50'S CHRISTMAS 307.25 115

## nCustomer nPurchase

## 201 7 7

## 343 1 1

## 153 1 1

## 431 2 2

## 454 2 2

## 135 5 5

head(cluster4[order(-cluster4$nPurchase),] )

## StockCode Description sumAmount sumQuantity

## 201 22423 REGENCY CAKESTAND 3 TIER 941.40 84

## 16 20685 DOORMAT RED RETROSPOT 96.30 12

## 485 M Manual -5719.22 0

## 135 22086 PAPER CHAIN KIT 50'S CHRISTMAS 307.25 115

## 215 22501 PICNIC BASKET WICKER LARGE 169.15 17

## 403 47566 PARTY BUNTING 151.95 31

## nCustomer nPurchase

## 201 7 7

## 16 6 6

## 485 6 6

## 135 5 5

## 215 5 5

## 403 5 5

cluster5 <- ddply(RFM\_cluster5, .(StockCode,Description), summarize, sumAmount= sum(Amount), sumQuantity= sum(Quantity), nCustomer= length(unique(CustomerID)), nPurchase= length(unique(InvoiceNo)) )

head(cluster5[order(-cluster5$sumQuantity),] )

## StockCode Description sumAmount sumQuantity

## 699 23166 MEDIUM CERAMIC TOP STORAGE JAR 77183.60 74215

## 901 84826 ASSTD DESIGN 3D PAPER STICKERS 0.00 12540

## 951 85123A WHITE HANGING HEART T-LIGHT HOLDER 6333.35 2465

## 28 18007 ESSENTIAL BALM 3.5g TIN IN ENVELOPE 144.00 2400

## 917 84950 ASSORTED COLOUR T-LIGHT HOLDER 1003.20 1824

## 886 84568 GIRLS ALPHABET IRON ON PATCHES 305.28 1728

## nCustomer nPurchase

## 699 1 1

## 901 1 1

## 951 20 20

## 28 1 1

## 917 1 1

## 886 2 2

head(cluster5[order(-cluster5$nCustomer),] )

## StockCode Description sumAmount sumQuantity

## 406 22423 REGENCY CAKESTAND 3 TIER 1953.15 169

## 951 85123A WHITE HANGING HEART T-LIGHT HOLDER 6333.35 2465

## 822 47566 PARTY BUNTING 875.35 193

## 988 M Manual 362.80 126

## 638 22960 JAM MAKING SET WITH JARS 250.75 59

## 70 21034 REX CASH+CARRY JUMBO SHOPPER 14.25 15

## nCustomer nPurchase

## 406 29 29

## 951 20 20

## 822 15 15

## 988 14 14

## 638 13 13

## 70 11 11

head(cluster5[order(-cluster5$sumAmount),] )

## StockCode Description sumAmount sumQuantity

## 699 23166 MEDIUM CERAMIC TOP STORAGE JAR 77183.60 74215

## 951 85123A WHITE HANGING HEART T-LIGHT HOLDER 6333.35 2465

## 400 22413 METAL SIGN TAKE IT OR LEAVE IT 3861.00 1404

## 406 22423 REGENCY CAKESTAND 3 TIER 1953.15 169

## 955 85152 HAND OVER THE CHOCOLATE SIGN 1192.50 621

## 917 84950 ASSORTED COLOUR T-LIGHT HOLDER 1003.20 1824

## nCustomer nPurchase

## 699 1 1

## 951 20 20

## 400 1 1

## 406 29 29

## 955 2 2

## 917 1 1

head(cluster5[order(-cluster5$nPurchase),] )

## StockCode Description sumAmount sumQuantity

## 406 22423 REGENCY CAKESTAND 3 TIER 1953.15 169

## 951 85123A WHITE HANGING HEART T-LIGHT HOLDER 6333.35 2465

## 822 47566 PARTY BUNTING 875.35 193

## 988 M Manual 362.80 126

## 638 22960 JAM MAKING SET WITH JARS 250.75 59

## 70 21034 REX CASH+CARRY JUMBO SHOPPER 14.25 15

## nCustomer nPurchase

## 406 29 29

## 951 20 20

## 822 15 15

## 988 14 14

## 638 13 13

## 70 11 11

Preprocessing: