# RFM Analysis Using SQLITE and Python

The RFM Model of customer value uses proven marketing principles to help businesses differentiate between marketing to existing and new customers and helps them create relevant and personalized messaging by understanding user behavior. The model allows the business to segment its customers based on three criteria based on an existing customer’s transaction history, namely:

Recency (When was the last time your customer purchased a product/service?)

A high recency score means a customer has positively considered your brand for a purchase decision recently. Recency can be scored by grading on custom-built filters such as bought on the last 7 days/1 month/3 months and so on, depending on the nature of the business.

Frequency (How often did the customer purchase in a year/fixed time period?)

A high-frequency score means a customer buys your brand frequently and is likely to be a loyalist of your brand. To calculate frequency, businesses need to analyze the total number of purchases completed by customers in a fixed time period. Frequency can be scored by grading on custom-built filters such as bought thrice in a year/bought once a month and so on, depending on the nature of the business.

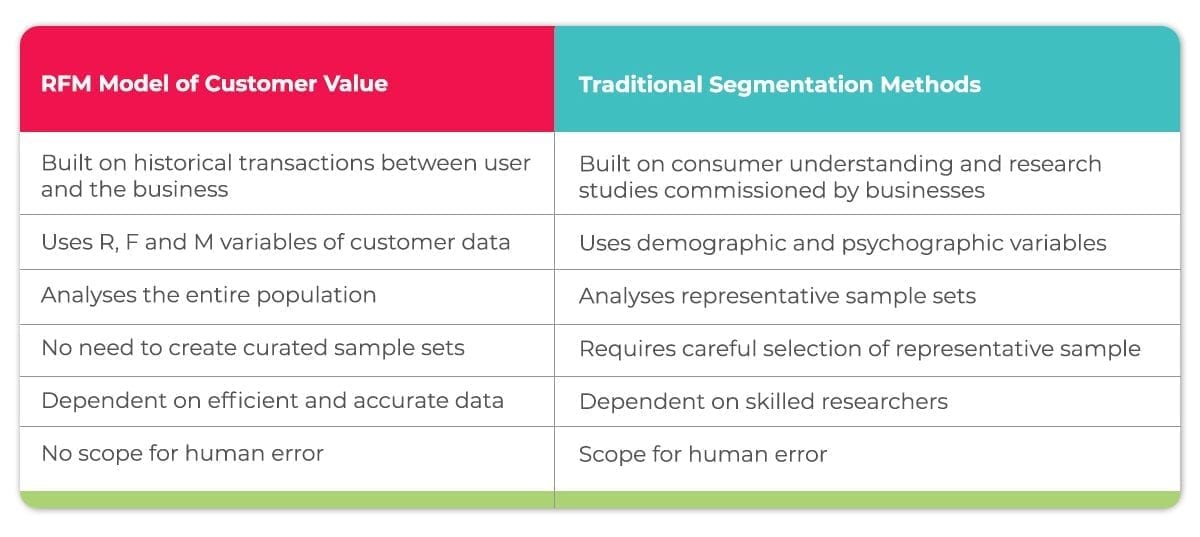
Monetary Value (How much money has the customer spent on your brand so far?)

A high monetary value score means a customer is one of the highest spending customers of your brand. Monetary value score can be graded on custom-built filters like spent more than Rs.10,000/30,000/50,0000 and so on, depending on the nature of the business.

All the above criteria can be graded on a scale of 1 to 5, with 5 being the best score you could assign a customer. It is also critical to specify an appropriate range for each grade, in order to create groupings of customers with similar buying behavior.

## Why is RFM better than traditional segmentation methods?

The RFM model is built on transactions between the user and the business, to create a robust data-backed method based on hard numbers. This customer data is graded, further analyzed, and then segmented in order to engage customers as distinct groups. This model helps businesses effectively analyze the past buying behavior of each customer, to predict and shape future customer interactions.



 Traditional methods of segmentation, used by market research companies before the advent of data analytics, use variables like demographic and psychographic factors to group their customers. Researchers always utilize sample audiences to predict population behavior, which reduces market researchers’ ability to predict user behavior of niche consumer sets and specific customers.

These studies are carried out manually, are dependent on skilled researchers, and are prone to human error. A sample could be incorrect, due to many reasons like an insufficient number of consumers, incorrect gender balance, varying psychographic factors, etc. These problems cannot occur in RFM, as it is a fundamentally data-centric model which analyses the entire population set, instead of a curated sample set. In addition to that, the variables of the RFM model are 100% accurate and precise, whereas traditional research involved factors like psychographics, which could be interpreted subjectively.

Using the RFM model helps a business define interactions with each specific customer, creating opportunities to increase the relevance of messaging, eventually creating the potential for increased customer lifetime value. RFM has the potential to create seamless interactions with high customer satisfaction, helping customers feel that the brand understands them and can effectively cater to their needs at all times.

## How RFM helps improve business understanding

RFM modeling increases a business’ ability to prevent churn by using fundamental marketing principles of segmentation, targeting, and positioning, which help understand the following:

Segmentation allows you to divide potential customers into distinct groups allowing businesses to talk to them separately. It helps answer the questions:

* *Are all my customers similar?*
* *What differentiates them from each other?*
* *Who is my most likely customer?*

Targeting involves understanding the routines and user behavior of these segments, allowing you to consider and choose the ideal way to speak to them. It helps answer the questions:

* *Where do my customers interact with the brand?*
* *What’s the best time, place, medium, and format to talk to them about my brand?*

Positioning helps you understand how to talk about your product/service, in order to maximize customer lifetime value. It helps answer the question:

* *What type of brand message will increase and ensure brand trust?*
* *What type of brand message is likely to induce a purchase interaction?*

Principles of segmentation, targeting, and positioning have been used for ages in the field of marketing. However, with the advent of data analytics, and the creation of number-driven models like RFM, the scope of these principles has widened tremendously. Today, businesses can go beyond the above questions with the help of the RFM model and get answers to highly specific questions such as:

* *Who are my best customers?*
* *Which customer has the potential to buy more?*
* *Which customer has been churned out/has lapsed?*
* *Which customer can the business afford to ignore to effectively utilize budgets?*
* *Which customer can be converted by creating value through promotions?*
* *Which customer is likely to be loyal in the near future?*

The RFM model allows businesses to gain key customer insights, through convenient data collection, and frame business strategy with those insights at the heart of every decision. The model allows the business to gain perspective on what their brand means to the existing customers, helps businesses manage customer perceptions, and also translates positive sentiment into purchase opportunities.

Businesses can recognize critical customer segments like churn-risk users, and create a bespoke marketing plan, specifically designed to retain those customers. Simultaneously, a business can also use the RFM model to maximize the potential of active customers, by creating personalized messaging and customized offerings, making them feel like high-value customers.

## Example RFM analysis-segmentation in SQLITE/PYTHON

The RFM model is fundamentally built using principles of data-driven marketing. Data-driven marketing has fundamentally transformed how marketing works ever since its inception, as it allows the analysis of large sets of customer data like never before. This has led to increased accuracy in understanding customers and enhanced ability to creatively customize messaging. The rise of automation in marketing technology has led to increased granularity and personalization, leading to enhanced relevance of each brand message.

Unlike traditional method that using demographic data, the idea of RFM analysis is to segment customers by transaction data. This makes RFM more practical than the traditional method, also RFM can analyse the entire population that available in the transaction data history. However, not all features in the data set will be used in RFM analysis, we only use these three features:

* The last time customer purchased a product/service (Recency)
* The number of purchases made by a customer during a certain period of time (Frequency)
* The amount of money spent by a customer during a certain period of time (Monetary)

In this article, I will do RFM analysis using SQLITE, the data set is an online retail data set available BB. The table contains 8 attributes as follow: (**Q: Which attributes represent dimensions?**)

1. ReceiptNo : Invoice number. Nominal, a 6-digit integral number that uniquely assigned to each transaction. Any number that starts with letter ‘c’ indicates a cancellation.
2. ProductCode : Product code. Nominal, a 5-digit integral number uniquely assigned to each distinct product.
3. Description : Product name. Nominal.
4. Quantity : The quantities of each product (item) per transaction. Numeric.
5. ReceiptDate : Invoice Date and time. Numeric, the day and time when each transaction was generated.
6. UnitPrice : Numeric, product price per unit in sterling.
7. CustomerID : Nominal, a 5-digit integral number uniquely assigned to each customer.
8. Country : Nominal, the name of the country where each customer resides.

**Data Pre-processing**

The first thing to do is create the table and import the data to Oracle.

|  |
| --- |
|  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**#**Set up and Read the data

import pandas as pd

import sqlite3

# read csv file

df\_Fact = pd.read\_csv('/Users/user/Desktop/dwh&BI/data.csv')

# connect to database

conn = sqlite3.connect("RFMdb")

cur = conn.cursor()

# load CRM data into the RFMdb database

df\_Fact.to\_sql("data", conn)

# CHECK IF DATA INSERTED INTO data/RFMdb

df\_Fact = pd.read\_sql('SELECT \* FROM data', conn)

df\_Fact

**Data include cancelled transactions, therefore any Receiptno containing ‘C’, NULLs in Customerid and 0 in UnitPrice must be omitted, using the following SQL statement. Also segment transactions on customer-id**

# clean data and group transactions by Customerid

cleandata= pd.read\_sql(''' SELECT customerid,

MAX(receiptdate) AS last\_order\_date,

COUNT(\*) AS count\_order,

SUM(unitprice\*quantity) AS totalprice

FROM data

WHERE receiptno NOT LIKE '%C%'

AND customerid IS NOT NULL

AND unitprice != 0

GROUP BY customerid ''', conn)

# Write clean CRM data into the database

cleandata.to\_sql("cleandata", conn)

**Calculate RFM Values Using SQL/SQLITE**

Divide the customer list into tiered groups for each of the three dimensions (R, F and M) Unless using specialized software, it’s recommended to divide the customers into four tiers for each dimension, such that each customer will be assigned to one tier in each dimension:

|  |  |  |
| --- | --- | --- |
| **Recency** | **Frequency** | **Monetary** |
| R-Tier-1 (most recent) | F-Tier-1 (most frequent) | M-Tier-1 (highest spend) |
| R-Tier-2 | F-Tier-2 | M-Tier-2 |
| R-Tier-3 | F-Tier-3 | M-Tier-3 |
| R-Tier-4 (least recent) | F-Tier-4 (only one transaction) | M-Tier-4 (low |

This results in 64 distinct customer segments (4x4x4), into which customers will be segmented. Three tiers can also be used (resulting in 27 segments); using more than four, however, is not recommended (because the difficulty in use outweighs the small benefit gain from the extra granularity).

#clean the data and calculate rfm values

df\_Fact = pd.read\_sql(''' SELECT customerid, rfm\_recency, rfm\_frequency, rfm\_monetary, rfm\_recency\*100 + rfm\_frequency\*10 + rfm\_monetary AS rfm\_combined

FROM

( SELECT customerid,

NTILE(4) OVER (ORDER BY last\_order\_date) AS rfm\_recency,

NTILE(4) OVER (ORDER BY count\_order) AS rfm\_frequency,

NTILE(4) OVER (ORDER BY totalprice) AS rfm\_monetary

FROM

cleandata

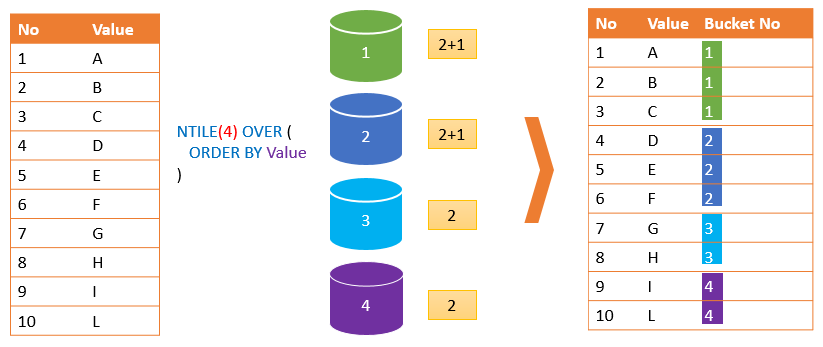
)

''', conn)

**There are 4338 numbers of customers segmented to one of 64 groups (4x4x4). Now I’d like to create a tree map for visualization, but I’ll do it in Python. Before that, it’s crucial to define the groups into some levels because having many groups is not practical.**

The NTILE() function assign numbers from 1 through the value of the expression to each row. The number of rows in buckets can differ by at most 1. The remainder of the number of rows divided by buckets is allocated to each bucket, starting with bucket 1.

For example, if you have 10 rows and 4 buckets. Each bucket will have 2 rows. The remainder of 10/4 is 2. So the first bucket will have 2 + 1 = 3 rows, the second bucket will also have 3 rows. The following picture illustrates the idea:



If the value of the expression is greater than the number of rows, then the NTILE() function will fill the number of buckets equal to the number of rows. Hence, the remaining buckets will be empty.

For example, if you have 10 rows and 11 buckets, each row will be assigned a bucket number from 1 to 10. The 11th bucket will have no row.

Notice that you cannot use a subquery or a window function in the expression

**Now I’d like to create a tree map for visualization, but I’ll do it in Python. Before that, it’s crucial to define the groups into some levels because having many groups is not practical. Here’s a table that explains how you can create up to 11 potential customer segments based on RFM scores.**

|  |  |  |
| --- | --- | --- |
| **Customer Segment** | **Recency Score Range** | **Frequency & Monetary Combined Score Range** |
| **Champions** | 4-5 | 4-5 |
| **Loyal Customers** | 2-5 | 3-5 |
| **Potential Loyalist** | 3-5 | 1-3 |
| **Recent Customers** | 4-5 | 0-1 |
| **Promising** | 3-4 | 0-1 |
| **Customers Needing Attention** | 2-3 | 2-3 |
| **About To Sleep** | 2-3 | 0-2 |
| **At Risk** | 0-2 | 2-5 |
| **Can’t Lose Them** | 0-1 | 4-5 |
| **Hibernating** | 1-2 | 1-2 |
| **Lost** | 0-2 | 0-2 |

**Marketers should assemble groups of customers most relevant for their particular business objectives and retention goals.**

**The RFM model is linked with the famous Pareto Principle, which says that 80% of total results are driven by the top 20% causes. When applied to marketing, it means that 80% of your total sales are likely to come from your top 20% of customers.**

import pandas as pd

import matplotlib.pyplot as plt

def rfm\_level(df\_Fact):

if ((df\_Fact['rfm\_recency'] >= 4) and (df\_Fact['rfm\_frequency'] >= 4)

and (df\_Fact['rfm\_monetary'] >= 4)):

return 'Best Customers'

elif ((df\_Fact['rfm\_recency'] >= 3) and (df\_Fact['rfm\_frequency'] >= 3)

and (df\_Fact['rfm\_monetary'] >= 3)):

return 'Loyal'

elif ((df\_Fact['rfm\_recency'] >= 3) and (df\_Fact['rfm\_frequency'] >= 1)

and (df\_Fact['rfm\_monetary'] >= 2)):

return 'Potential Loyalist'

elif ((df\_Fact['rfm\_recency'] >= 3) and (df\_Fact['rfm\_frequency'] >= 1)

and (df\_Fact['rfm\_monetary'] >= 1)):

return 'Promising'

elif ((df\_Fact['rfm\_recency'] >= 2) and (df\_Fact['rfm\_frequency'] >= 2)

and (df\_Fact['rfm\_monetary'] >= 2)):

return 'Customers Needing Attention'

elif ((df\_Fact['rfm\_recency'] >= 1) and (df\_Fact['rfm\_frequency'] >= 2)

and (df\_Fact['rfm\_monetary'] >= 2)):

return 'At Risk'

elif ((df\_Fact['rfm\_recency'] >= 1) and (df\_Fact['rfm\_frequency'] >= 1)

and (df\_Fact['rfm\_monetary'] >= 2)):

return 'Hibernating'

else:

return 'Lost'

#Create a new variable rfm\_level

df\_Fact['rfm\_level'] = df\_Fact.apply(rfm\_level, axis=1)

df\_Fact

|  | **CustID** | **R** | **F** | **M** | **COMB** | **rfm\_level** |
| --- | --- | --- | --- | --- | --- | --- |
| **0** | 16282 | 1 | 1 | 1 | 111 | Lost |
| **1** | 15266 | 1 | 1 | 1 | 111 | Lost |
| **2** | 15464 | 1 | 4 | 3 | 143 | At Risk |
| **3** | 13187 | 1 | 2 | 1 | 121 | Lost |
| **4** | 12501 | 1 | 4 | 4 | 144 | At Risk |
| **...** |  | ... | ... | ... | ... | ... |
| **4333** | 13813 | 4 | 3 | 3 | 433 | Loyal |
| **4334** | 12676 | 4 | 3 | 3 | 433 | Loyal |
| **4335** | 15602 | 4 | 3 | 3 | 433 | Loyal |
| **4336** | 15321 | 4 | 3 | 4 | 434 | Loyal |
| **4337** | 17725 | 4 | 4 | 4 | 444 | Best Customers |

# Calculate total customers in each segment

rfm\_agg = df\_Fact.groupby('rfm\_level').agg({'customerid':'count'})

print(rfm\_agg)

customerid

rfm\_level

At Risk 475

Best Customers 414

Customers Needing Attention 665

Hibernating 211

Lost 819

Loyal 907

Potential Loyalist 581

Promising 266

#RFM visualization, you may have to install squarify

import squarify

fig = plt.gcf()

ax = fig.add\_subplot()

fig.set\_size\_inches(13, 7)

squarify.plot(sizes=rfm\_agg['customerid '],

label=['At Risk',

'Best Customers',

'Customers Needing Attention',

'Hibernating',

'Lost',

'Loyal',

'Potential Loyalist',

'Promising'], alpha=0.7)

plt.title("RFM Segments",fontsize=20)

plt.axis('off')

plt.show()

Chart, treemap chart



**How to install squarify on Anaconda**

* Open Anaconda Powershell Prompt (APP)
* To install this package with conda type:

conda install -c conda-forge squarify

on (APP) and prese Enter….its all done for you!!

* Or simply! pip install squarify

The End.