Customer Segmentation with RFM and Clustering

Joseph Bae
DTSA-5510 Final Project

Customer Segmentation

- Customer Segmentation is dividing a customer base into groups based on purchase behavior
- Allows companies to understand their customers and how to market to their different customer types
- Example segments:
  - Loyal/Frequent customers
  - Once-a-year purchasers
  - Inactive customers (haven't bought in a while)

shutterstock.com · 1750056179

RFM for Customer Segmentation

- Customers are analyzed based on recency, frequency, and monetary purchase behavior.
- They are segmented, and grouped, based on their RFM values.

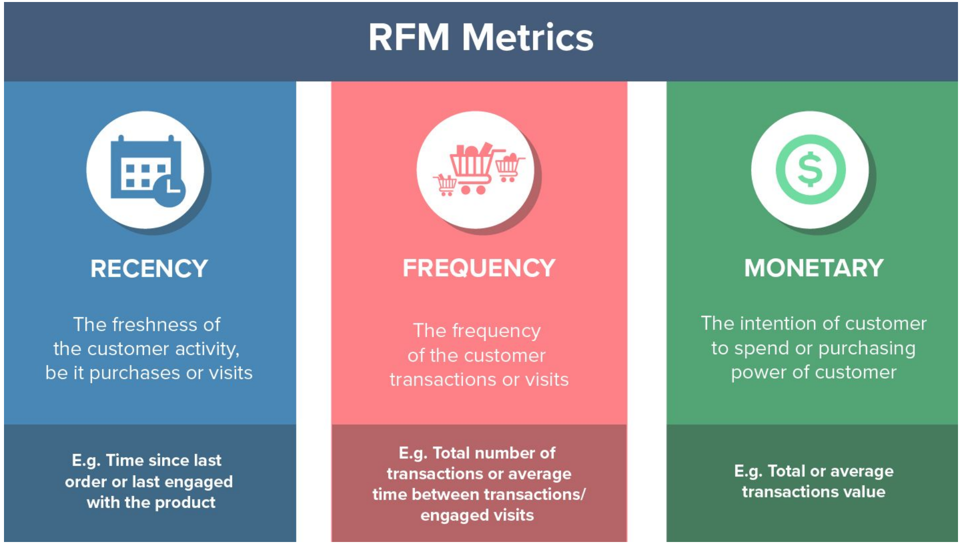


Image source: https://ordorite.com/how-customer-segmentation-can-improve-your-profits/

Dataset

Joseph Bae
DTSA-5510 Final Project

Online Retail Dataset (UCI ML Repository)

Online Retail

Donated on 11/5/2015

This is a transnational data set which contains all the transactions occurring between 01/12/2010 and 09/12/2011 for a UK-based and registered non-store online retail.

Dataset Characteristics
Multivariate, Sequential, Time-Series

Subject Area
Business

Associated Tasks
Classification, Clustering

Feature Type
Integer, Real

 # Instances
541909

 # Features
6

https://archive.ics.uci.edu/dataset/352/online+retail

Project Objectives

Joseph Bae
DTSA-5510 Final Project

Objective

Step 1
EDA

- Understand features
- Charts, histograms
- Check data quality
- Check missing data
- Feature correlation

Step 2
RFM w/ Quartiles

- Compute RFM values
- Compute quartiles
- Group customers by quartile

Step 3
RFM w/ Clustering

- Use same RFM values
- K-Means Clustering
- Hierarchical Clustering

Step 4
Comparison

- Quartiles vs Clustering
- Pros/Cons from data and business perspective

EDA

Joseph Bae
DTSA-5510 Final Project

EDA Findings

- 541,909 data points
- 8 features

#	Column	Non-Null Count	Dtype
---	------	--------------	-----
0	InvoiceNo	541909 non-null	object
1	StockCode	541909 non-null	object
2	Description	540455 non-null	object
3	Quantity	541909 non-null	int64
4	InvoiceDate	541909 non-null	datetime64[ns]
5	UnitPrice	541909 non-null	float64
6	CustomerID	406829 non-null	float64
7	Country	541909 non-null	object

EDA Findings

```
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   InvoiceNo    541909 non-null  object
 1   StockCode    541909 non-null  object
 2   Description  540455 non-null  object
 3   Quantity     541909 non-null  int64
 4   InvoiceDate  541909 non-null  datetime64[ns]
 5   UnitPrice    541909 non-null  float64
 6   CustomerID   406829 non-null  float64
 7   Country      541909 non-null  object
```

InvoiceNo

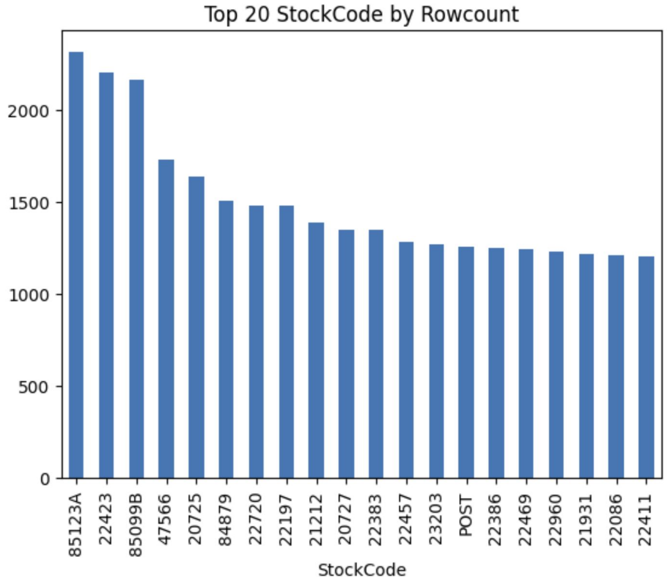
- Identifies customer purchases
- Cancelled invoices begin with ‘C’, otherwise are numeric
- Multiple rows can have same InvoiceNo

EDA Findings

```
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   InvoiceNo    541909 non-null  object
 1   StockCode    541909 non-null  object
 2   Description  540455 non-null  object
 3   Quantity     541909 non-null  int64
 4   InvoiceDate  541909 non-null  datetime64[ns]
 5   UnitPrice    541909 non-null  float64
 6   CustomerID   406829 non-null  float64
 7   Country      541909 non-null  object
```

StockCode

- Identifies products
- Some StockCode have over 2000 rows



EDA Findings

```
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   InvoiceNo    541909 non-null  object
 1   StockCode    541909 non-null  object
 2   Description  540455 non-null  object
 3   Quantity     541909 non-null  int64
 4   InvoiceDate  541909 non-null  datetime64[ns]
 5   UnitPrice    541909 non-null  float64
 6   CustomerID   406829 non-null  float64
 7   Country      541909 non-null  object
```

Description

- Most popular product: White Hanging Heart T-Light Holder shows up in 2,313 rows
- 1,454 rows with Null description, but they have StockCode
- Word cloud:

EDA Findings

```
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   InvoiceNo    541909 non-null  object
 1   StockCode    541909 non-null  object
 2   Description  540455 non-null  object
 3   Quantity     541909 non-null  int64
 4   InvoiceDate  541909 non-null  datetime64[ns]
 5   UnitPrice    541909 non-null  float64
 6   CustomerID   406829 non-null  float64
 7   Country      541909 non-null  object
```

Quantity

- Can be negative, for cancelled orders
- Total quantity per customer has large range of values, with some customer having negative quantity.

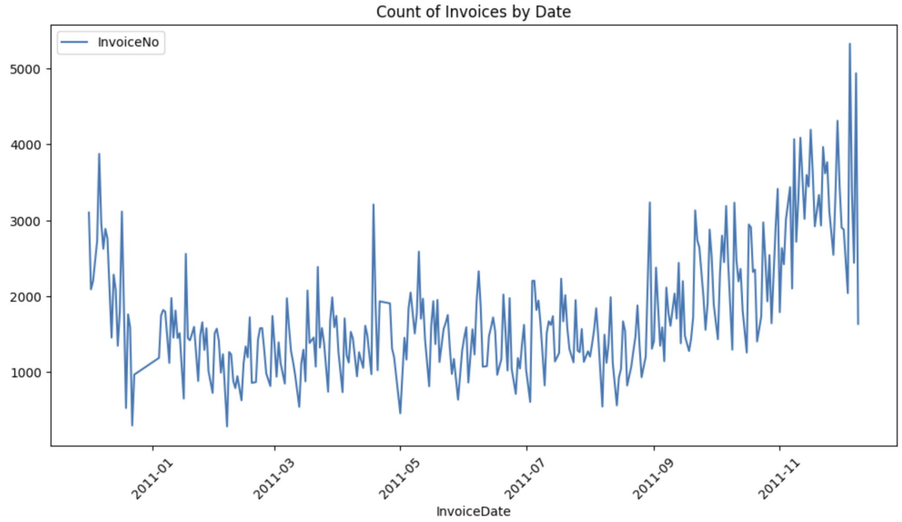
	CustomerID	Quantity	UnitPrice
3103	16546.0	-303	53.03
2578	15823.0	-283	85.19
1384	14213.0	-244	24.45
3245	16742.0	-189	472.65
2892	16252.0	-158	67.10
...	...	...	...
4233	18102.0	64122	5159.73
3758	17450.0	69029	3320.09
1895	14911.0	77180	31060.66
55	12415.0	77242	2499.82
1703	14646.0	196719	5400.21

EDA Findings

#	Column	Non-Null Count	Dtype
---	------	--------------	-----
0	InvoiceNo	541909 non-null	object
1	StockCode	541909 non-null	object
2	Description	540455 non-null	object
3	Quantity	541909 non-null	int64
4	InvoiceDate	541909 non-null	datetime64[ns]
5	UnitPrice	541909 non-null	float64
6	CustomerID	406829 non-null	float64
7	Country	541909 non-null	object

InvoiceDate

- Ranges from Dec 2010 to Dec 2011:



EDA Findings

```
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   InvoiceNo    541909 non-null  object
 1   StockCode    541909 non-null  object
 2   Description  540455 non-null  object
 3   Quantity     541909 non-null  int64
 4   InvoiceDate  541909 non-null  datetime64[ns]
 5   UnitPrice    541909 non-null  float64
 6   CustomerID   406829 non-null  float64
 7   Country      541909 non-null  object
```

UnitPrice

- Has a mean of 4.59 but ranges from -11K to 39K

```
count    541718.000000
mean          4.591659
std          96.548583
min      -11062.060000
25%           1.250000
50%           2.080000
75%           4.130000
max       38970.000000
Name: UnitPrice, dtype: float64
```

- Product descriptions with UnitPrice above 500 or below 0 are removed (except for PICNIC BASKET...)

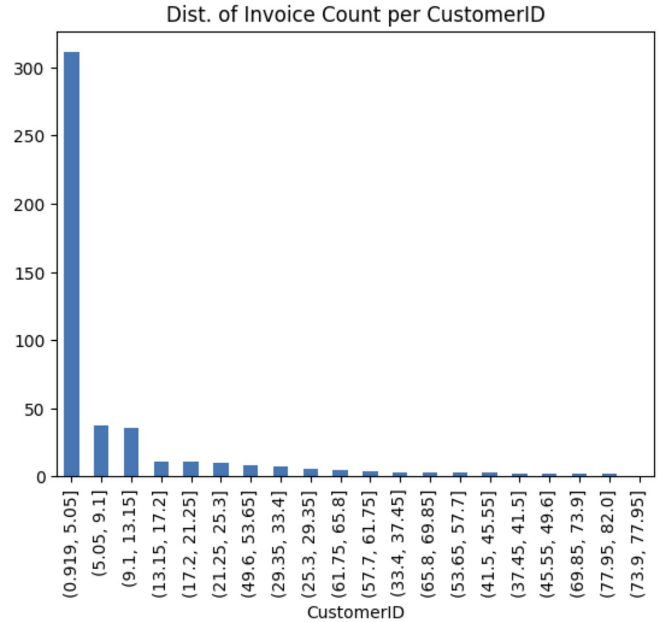
```
Description
DOTCOM POSTAGE                  105
Manual                           92
AMAZON FEE                       31
CRUK Commission                   7
POSTAGE                           6
Bank Charges                      4
Adjust bad debt                   3
PICNIC BASKET WICKER 60 PIECES    2
Discount                          1
SAMPLES                           1
Name: count, dtype: int64
```

EDA Findings

#	Column	Non-Null Count	Dtype
0	InvoiceNo	541909 non-null	object
1	StockCode	541909 non-null	object
2	Description	540455 non-null	object
3	Quantity	541909 non-null	int64
4	InvoiceDate	541909 non-null	datetime64[ns]
5	UnitPrice	541909 non-null	float64
6	CustomerID	406829 non-null	float64
7	Country	541909 non-null	object

CustomerID

- Has 134K null entries, which were removed
- Distribution of Invoices per Customer is right-skewed, with the vast majority having between 1 and 5 orders.

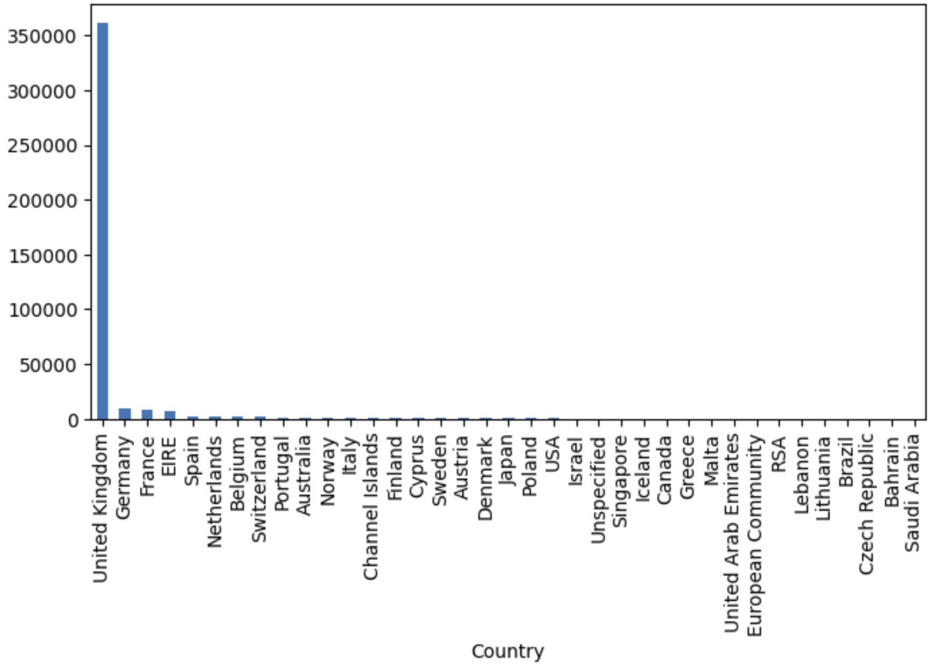


EDA Findings

#	Column	Non-Null Count	Dtype
0	InvoiceNo	541909 non-null	object
1	StockCode	541909 non-null	object
2	Description	540455 non-null	object
3	Quantity	541909 non-null	int64
4	InvoiceDate	541909 non-null	datetime64[ns]
5	UnitPrice	541909 non-null	float64
6	CustomerID	406829 non-null	float64
7	Country	541909 non-null	object

Country

- Almost all data points are for UK purchases



EDA Findings

Numerical representation of InvoiceDate

Correlation

- Numerical features: InvoiceDateNum, UnitPrice, Quantity
- There's very little correlation between these numerical features

RFM w/ Quartiles (non-ML Approach)

Joseph Bae
DTSA-5510 Final Project

RFM for Customer Segmentation w/ Quartiles

- RFM values computed as:
  - Recency = # days since last order (excl. cancelled)
  - Frequency = # unique invoices
  - Monetary = Total (UnitPrice)*(Quantity)
- Quartiles approach - Bin RFM values into quartiles, and customers are scored 1-4 for R, F, and M, producing 4*4*4 = 64 max possible segments.

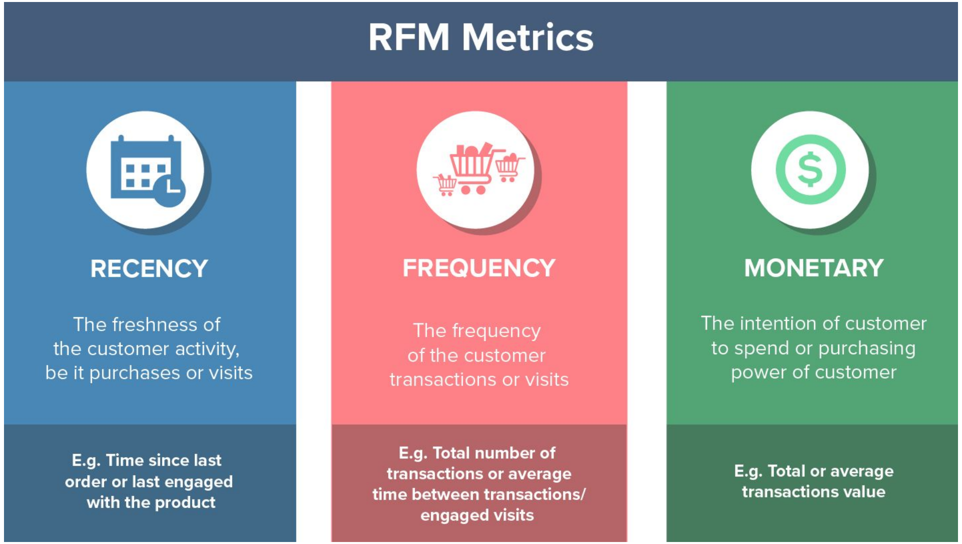
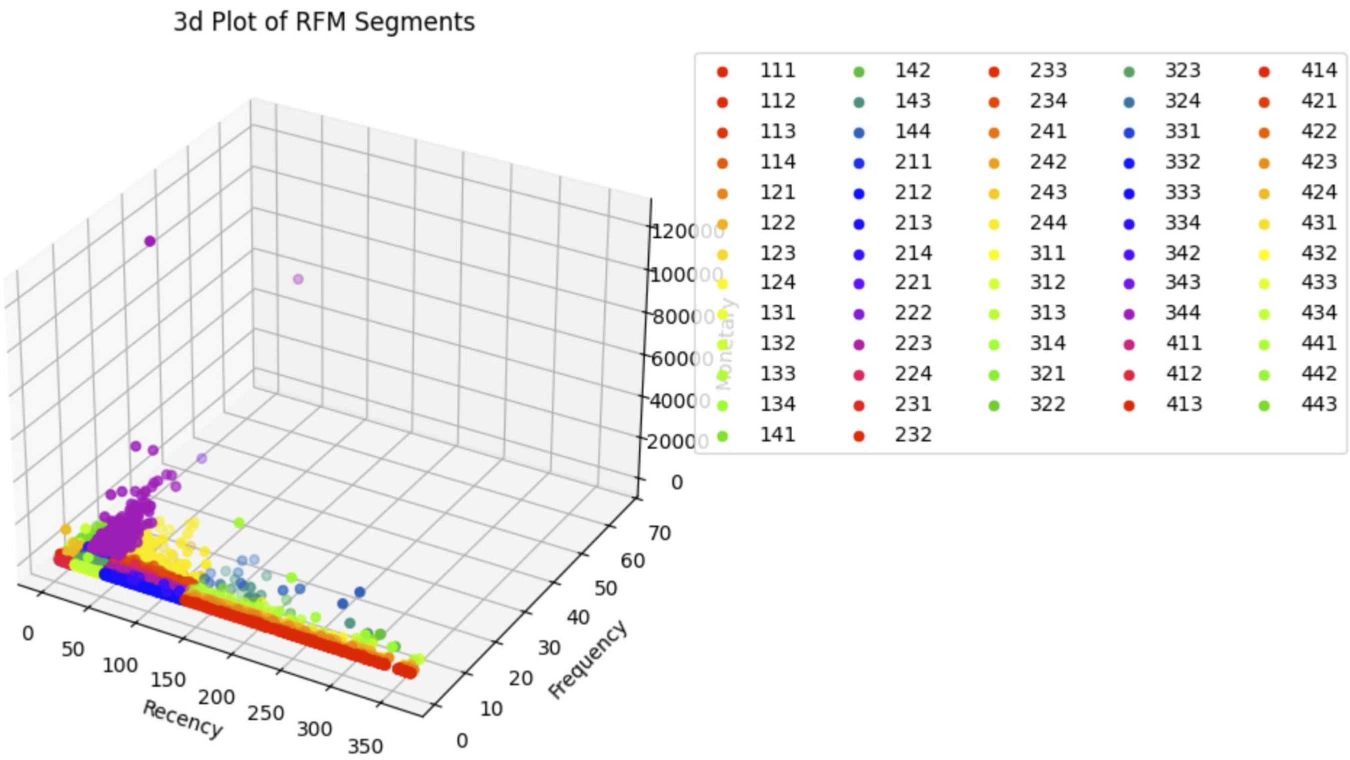


Image source: https://ordorite.com/how-customer-segmentation-can-improve-your-profits/

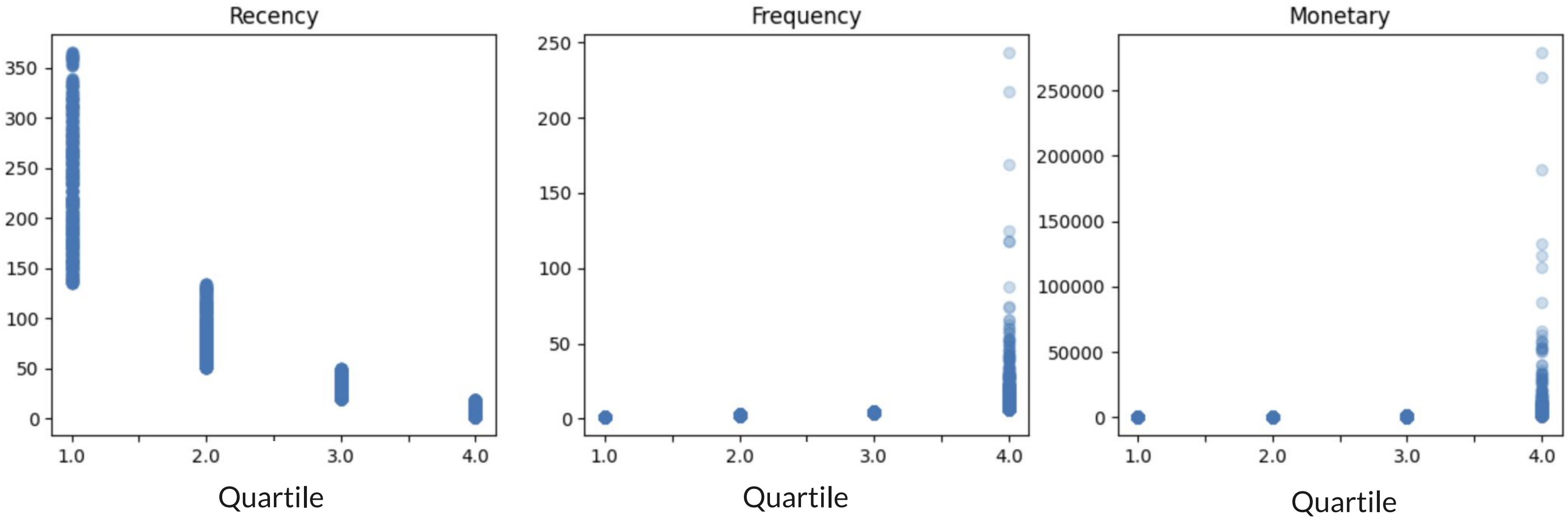
RFM for Customer Segmentation w/ Quartiles

- Using quartile approach, we get 62 customer segments
- Segments need to be grouped together to be useful.
  - For ex. 8-12 larger segments



RFM for Customer Segmentation w/ Quartiles

- Using quartile approach we see that quartiles 1-3 for Frequency and Monetary have the same values, while quartile 4 has a huge range. To fix, need to tweak value ranges for the F and M bins until a more even distribution is reached.



Model Conclusions

01

RFM Segmentation w/ Quartiles (Non-ML Approach)

- 62 customer segments that need to be analyzed & grouped
- Frequency quartiles 1-3 all have the same frequency values
- Monetary quartiles 1-3 all have the same monetary values
- Needs further tweaking and planning

RFM w/ Clustering (ML Approach)

Joseph Bae
DTSA-5510 Final Project

RFM for Customer Segmentation w/ Clustering

- Idea: take same computed RFM values and use Clustering to create segments instead.
- No need to analyze and group 62 RFM segments together, clustering does the grouping.
- We'll use 2 clustering algorithms and compare results
- Important: Standardize the features (R, F, M values) before clustering so that they are treated equally important by the algorithms

K-Means Clustering

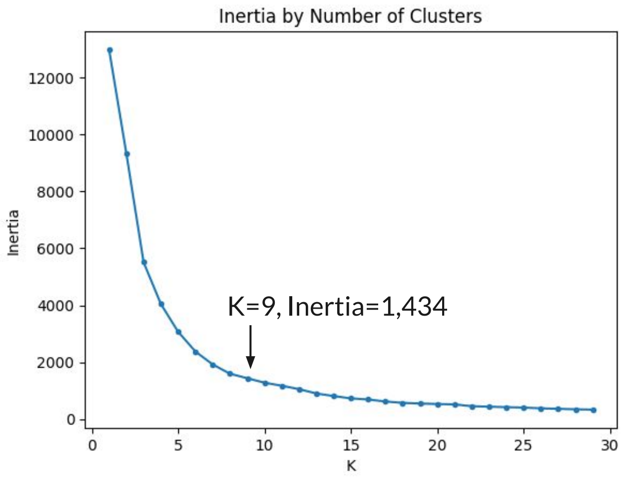
Hierarchical Clustering

K-Means Clustering

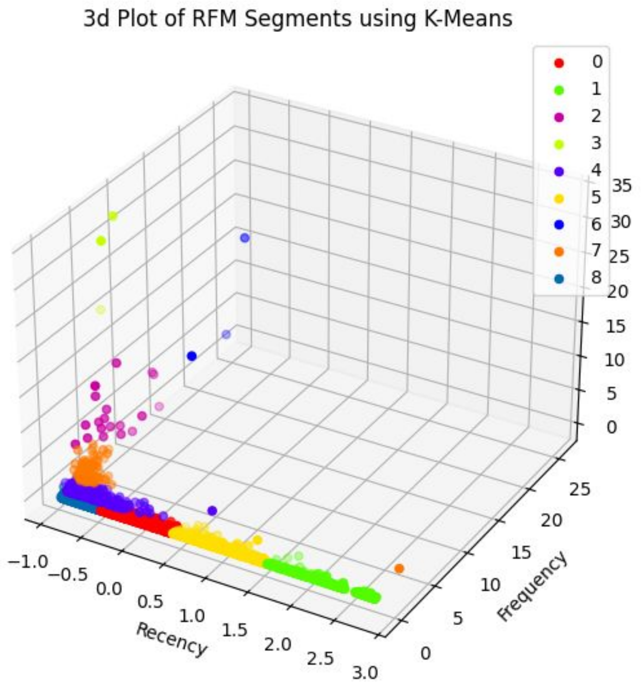
Joseph Bae
DTSA-5510 Final Project

RFM for Customer Segmentation w/ K-Means Clustering

- To determine how many clusters to use in K-Means, the WCSS (aka Inertia) was plotted against # of clusters (K).
- A la the 'Elbow Method', where the curve starts to straighten out is where our optimal number of clusters is, at K=9.

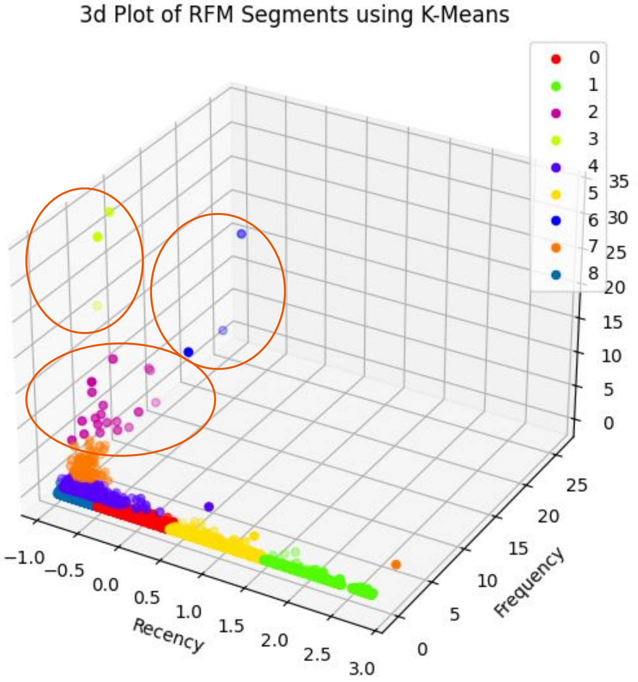


RFM for Customer Segmentation w/ K-Means Clustering



KM_Cluster	Size	Size%	Recency	Frequency	Monetary	Description
0	957	22%	Medium	Low	Low	Inactive Customers
1	501	12%	High	Low	Low	Lapsed Customers
2	19	0%	Low	Medium	Medium	Outliers (med freq., med monetary)
3	3	0%	Low	Medium	High	Outliers (high monetary)
4	532	12%	Low	Low	Low	Recently Active 2* Customers
5	605	14%	High	Low	Low	Almost Lapsed Customers
6	3	0%	Low	High	Medium	Outliers (high frequency)
7	112	3%	Low	Medium	Low	Recently Active 3* Customers
8	1599	37%	Low	Low	Low	Recently Active 1* Customers

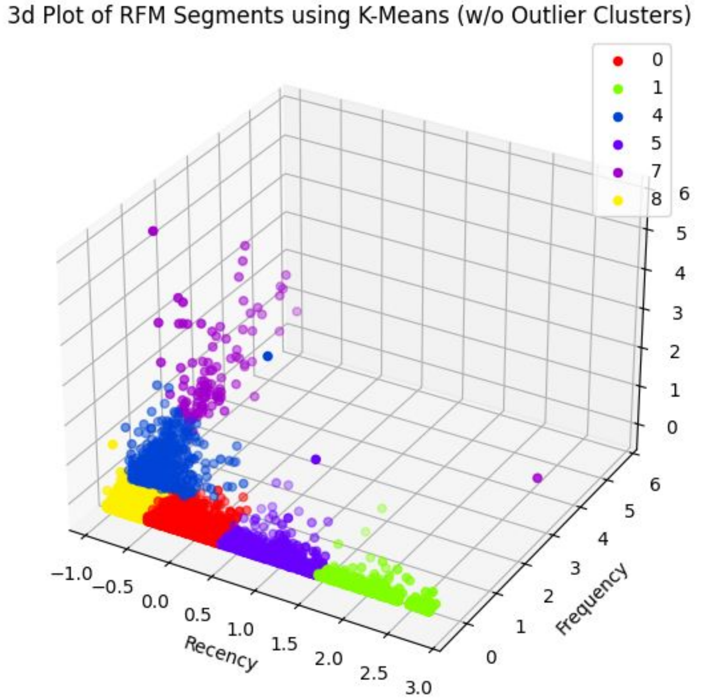
RFM for Customer Segmentation w/ K-Means Clustering



Ignore 'outlier' clusters that contain < 1% of customers..

KM_Cluster	Size	Size%	Recency	Frequency	Monetary	Description
0	957	22%	Medium	Low	Low	Inactive Customers
1	501	12%	High	Low	Low	Lapsed Customers
2	19	0%	Low	Medium	Medium	Outliers (med freq., med monetary)
3	3	0%	Low	Medium	High	Outliers (high monetary)
4	532	12%	Low	Low	Low	Recently Active 2* Customers
5	605	14%	High	Low	Low	Almost Lapsed Customers
6	3	0%	Low	High	Medium	Outliers (high frequency)
7	112	3%	Low	Medium	Low	Recently Active 3* Customers
8	1599	37%	Low	Low	Low	Recently Active 1* Customers

RFM for Customer Segmentation w/ K-Means Clustering



KM_Cluster	Size	Size%	Recency	Frequency	Monetary	Description
0	957	22%	Medium	Low	Low	Inactive Customers
1	501	12%	High	Low	Low	Lapsed Customers
4	532	12%	Low	Low	Low	Recently Active 2* Customers
5	605	14%	High	Low	Low	Almost Lapsed Customers
7	112	3%	Low	Medium	Low	Recently Active 3* Customers
8	1599	37%	Low	Low	Low	Recently Active 1* Customers

Model Conclusions

01

RFM Segmentation w/ Quartiles (Non-ML Approach)

- 62 customer segments that need to be analyzed & grouped
- Frequency quartiles 1-3 all have the same frequency values
- Monetary quartiles 1-3 all have the same monetary values
- Needs further tweaking and planning

02

RFM Segmentation w/ K-Means Clusters (9 Clusters)

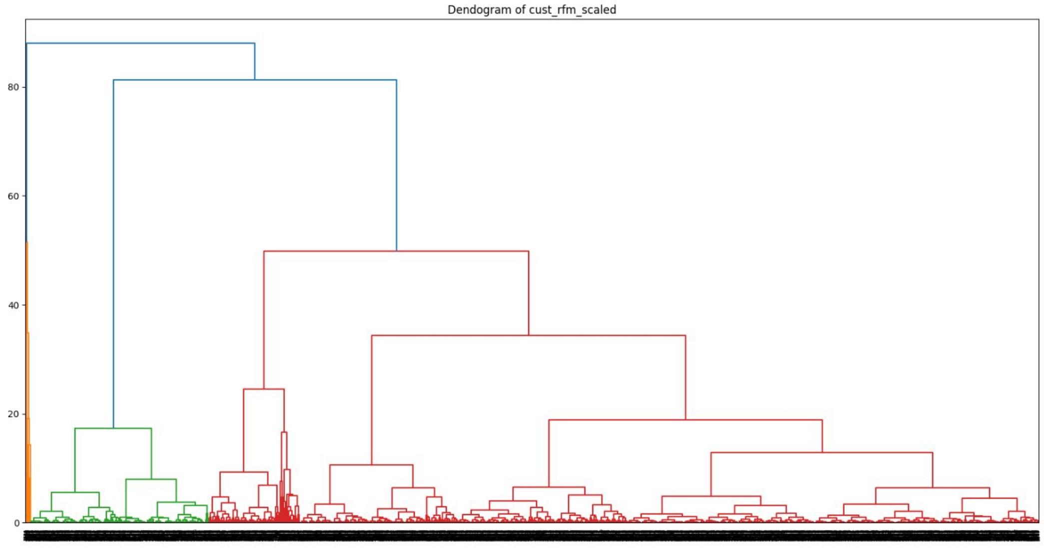
- 9 clusters, but 3 contain only outliers, so 6 are ‘useable’
- Useable clusters are well defined and have business significance
- Recency is divided into 4 levels - recently active, inactive, almost lapsed, lapsed
- Frequency+Monetary are divided into 3 levels - 1-star, 2-star, 3-star customers

Hierarchical Agglomerative Clustering

Joseph Bae
DTSA-5510 Final Project

RFM for Customer Segmentation w/ Hierarchical Clusters

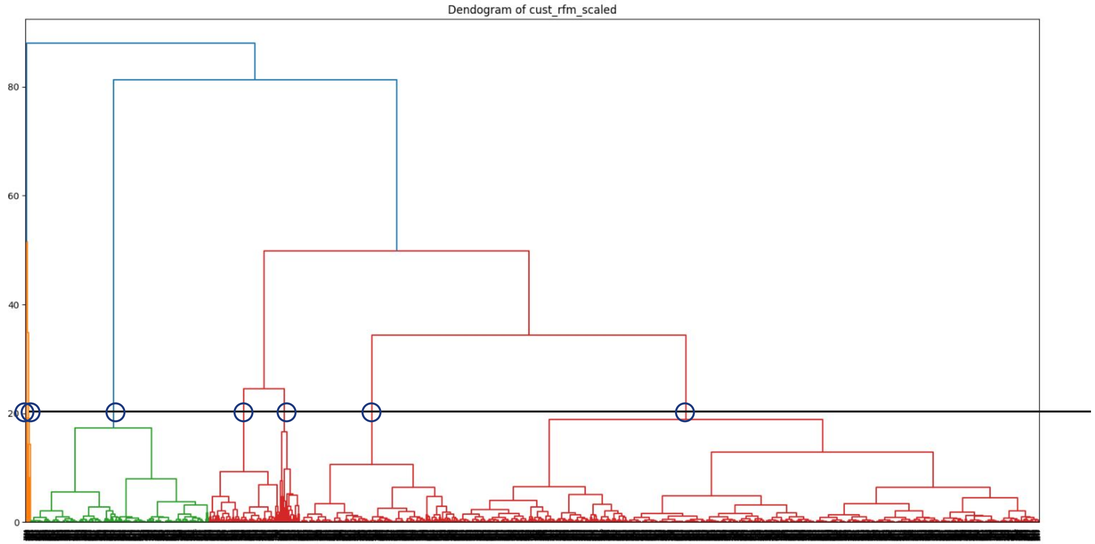
- To determine how many clusters to use for Agglomerative clustering, we visually inspect the dendrogram
- Draw a line at height where the vertical distances between clusters starts to become very large. Choosing height=20 gives us 7 clusters



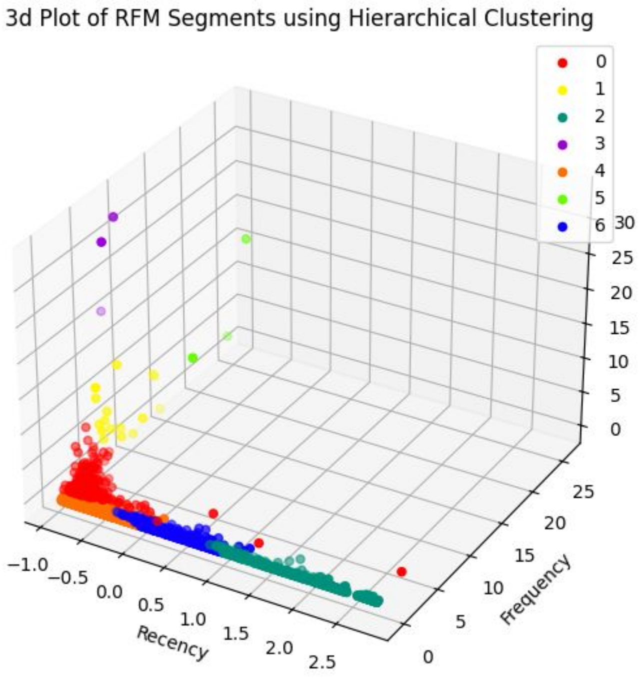
RFM for Customer Segmentation w/ Hierarchical Clusters

- To determine how many clusters to use for Agglomerative clustering, we visually inspect the dendrogram
- Draw a line at height where the vertical distances between clusters starts to become very large. Choosing height=20 gives us 7 clusters

Left lines are hard-to-spot due to outliers

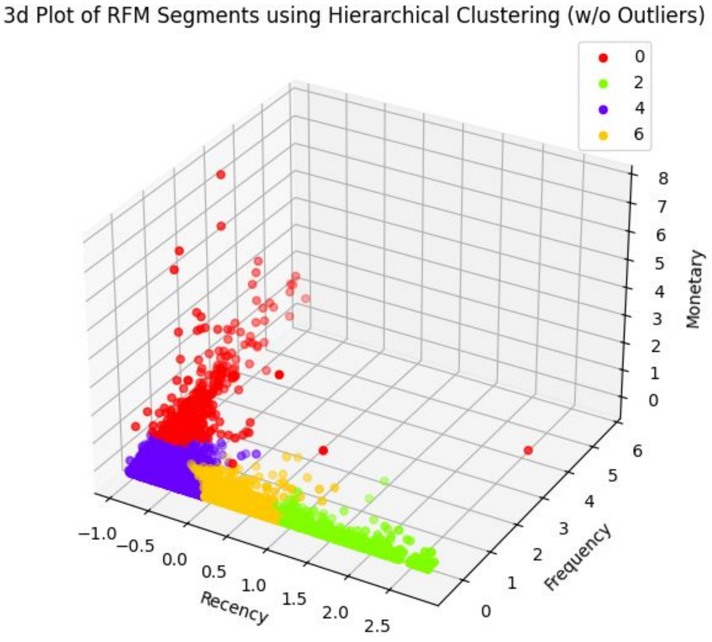


RFM for Customer Segmentation w/ Hierarchical Clusters



AC_Cluster	Size	Size%	Recency	Frequency	Monetary	Description
0	387	9%	Low	Low/Medium	Low/Medium	Recently Active 2-3* Customers
1	16	0%	Low	Medium	Medium	Outliers (med freq., med monetary)
2	763	18%	High	Low	Low	Lapsing & Lapsed Customers
3	3	0%	Low	Medium	High	Outliers (high monetary)
4	2480	57%	Low	Low	Low	Recently Active 1* Customers
5	3	0%	Low	High	Medium	Outliers (high frequency)
6	679	16%	Medium/High	Low	Low	Inactive Customers

RFM for Customer Segmentation w/ Hierarchical Clusters



AC_Cluster	Size	Size%	Recency	Frequency	Monetary	Description
0	387	9%	Low	Low/Medium	Low/Medium	Recently Active 2-3* Customers
2	763	18%	High	Low	Low	Lapsing & Lapsed Customers
4	2480	57%	Low	Low	Low	Recently Active 1* Customers
6	679	16%	Medium/High	Low	Low	Inactive Customers

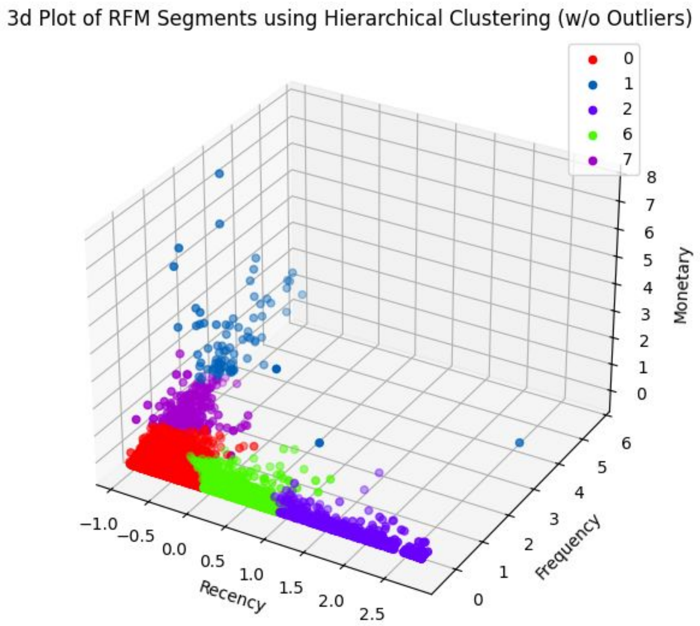
Model Conclusions

01	RFM with Quartiles (Non-ML Approach)	• 62 customer segments that need to be analyzed & grouped • Frequency quartiles 1-3 all have the same frequency values • Monetary quartiles 1-3 all have the same monetary values • Needs further tweaking and planning
02	RFM with K-Means Clustering (9 Clusters)	• 9 clusters, but 3 contain only outliers, so 6 are 'useable' • Useable clusters are well defined and have business significance • Recency is divided into 4 levels - recently active, inactive, almost lapsed, lapsed • Frequency+Monetary are divided into 3 levels - 1-star, 2-star, 3-star customers
03	RFM with Agglomerative Clustering (7 Clusters)	• Only 4 clusters are 'useable', since 3 only had outliers • Clusters are well-defined, but there may be too few of them • Previous 2-star & 3-star customers are combined, making it harder to define • Similarly, 2 recency 'levels' are combined • Long runtime (when producing dendrogram)

Hierarchical Agglomerative Clustering with 9 Clusters

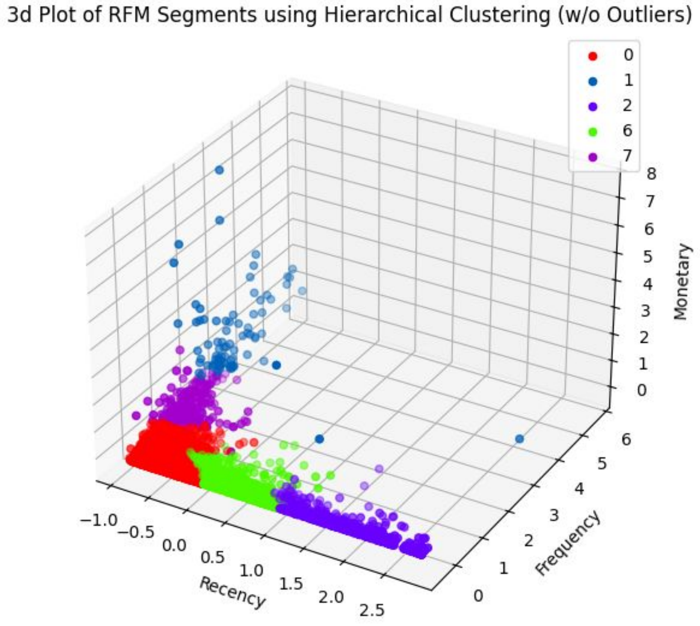
Joseph Bae
DTSA-5510 Final Project

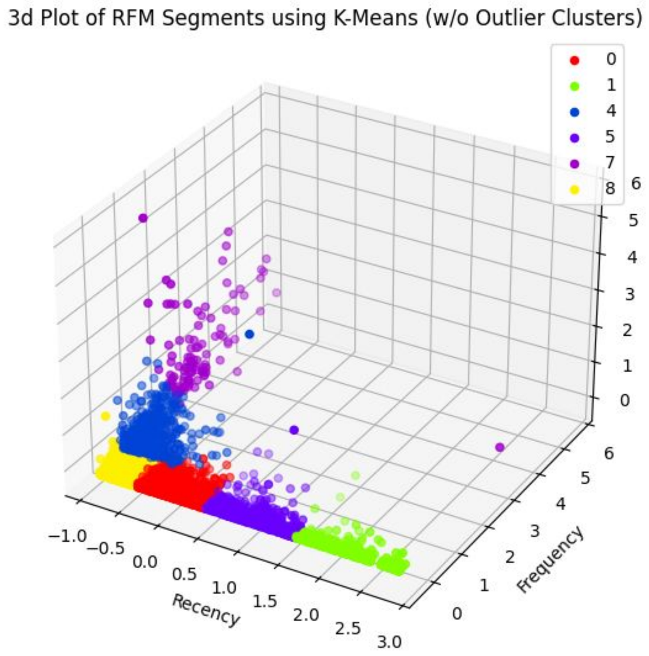
RFM w/ Hierarchical Clusters, 9 Clusters



- Agglomerative Clustering with K=9 had 4 outlier segments which are removed in left 3d plot.
- Having 1 additional cluster helps divide the non-outlier customers into more closely defined groups, but:
  - Cluster 0 is still quite large and varied
  - Recency is divided into 3 levels, instead of 4 like under K-Means
  - Cluster 1 (blue) is fairly small - only 84 customers

RFM w/ Hierarchical Clusters, 9 Clusters





Model Conclusions

01	RFM with Quartiles (Non-ML Approach)	• 62 customer segments that need to be analyzed & grouped • Frequency quartiles 1-3 all have the same frequency values • Monetary quartiles 1-3 all have the same monetary values • Needs further tweaking and planning
02	RFM with K-Means Clustering (9 Clusters)	• 9 clusters, but 3 contain only outliers, so 6 are ‘useable’ • Useable clusters are well defined and have business significance • Recency is divided into 4 levels - recently active, inactive, almost lapsed, lapsed • Frequency+Monetary are divided into 3 levels - 1-star, 2-star, 3-star customers
03	RFM with Agglomerative Clustering (7 Clusters)	• Only 4 clusters are ‘useable’, since 3 only had outliers • Clusters are well-defined, but have less business significance • Previous 2-star & 3-star customers are combined, making it harder to define • Similarly, 2 recency ‘levels’ are combined • Long runtime (when producing dendrogram)
04	RFM with Agglomerative Clustering (9 Clusters)	• 5 useable clusters, since 4 only had outliers • Clusters are better defined than with K=7, but still not as good as K-Means • 1 Cluster has a large variety of customers, and 1 cluster only has 84 members • Long runtime (when producing dendrogram)

Model Conclusions

01	RFM with Quartiles (Non-ML Approach)	• 62 customer segments that need to be analyzed & grouped • Frequency quartiles 1-3 all have the same frequency values • Monetary quartiles 1-3 all have the same monetary values • Needs further tweaking and planning
02	RFM with K-Means Clustering (9 Clusters)	• 9 clusters, but 3 contain only outliers, so 6 are ‘useable’ • Useable clusters are well defined and have business significance • Recency is divided into 4 levels - recently active, inactive, almost lapsed, lapsed • Frequency+Monetary are divided into 3 levels - 1-star, 2-star, 3-star customers
03	RFM with Agglomerative Clustering (7 Clusters)	• Only 4 clusters are ‘useable’, since 3 only had outliers • Clusters are well-defined, but have less business significance • Previous 2-star & 3-star customers are combined, making it harder to define • Similarly, 2 recency ‘levels’ are combined • Long runtime (when producing dendrogram)
04	RFM with Agglomerative Clustering (9 Clusters)	• 5 useable clusters, since 4 only had outliers • Clusters are better defined than with K=7, but still not as good as K-Means • 1 Cluster has a large variety of customers, and 1 cluster only has 84 members • Long runtime (when producing dendrogram)

Final Thoughts

Joseph Bae
DTSA-5510 Final Project

Final Thoughts

- The output from K-Means for this particular project was better, but both clustering methods produced clear segments. Factors that could have changed the outcome:
  - Removing outliers before clustering may significantly change resulting clusters
  - ‘Unfavorable’ initial set of centroids from K-Means due to randomization
- Hierarchical clustering - producing dendrogram had a long runtime for just 4.3K customers
- Quartile approach without ML is the most easy-to-explain, which justifies its popularity. However, results will take more fine-tuning and iterations to get the major customer segments, compared to clustering approaches which give that to us more instantly.

Links

- Github repository: https://github.com/jDyn90/dtsa5510