**Customer Segmentation Analysis**

Objective: segment customers based on groups based on their last purchase and number of purchases.

# DATA OVERVIEW

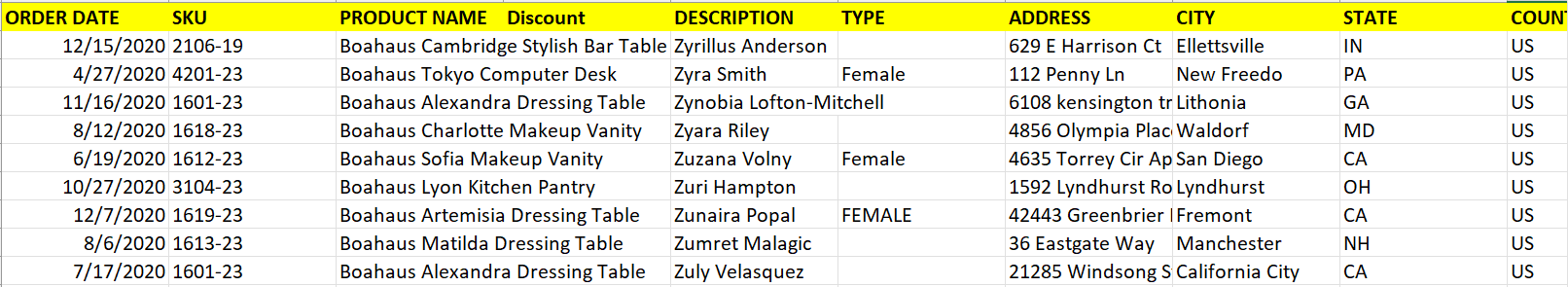
Dimensions: 91,380 x 12.

Data containing over 90k registers about furniture purchases.

## Key Columns

* **ORDER DATE**: Day of purchase.
* **DESCRIPTION**: Customer Name (can be a person or a corporation).
* **ZIP CODE**: Customer’s ZIP Code

*Dataframe name: df*



# DATA PRE-PROCESSING

Lowercased all customer names to avoid not spotting duplicates

Removed rows with null DESCRIPTION and ZIP CODE

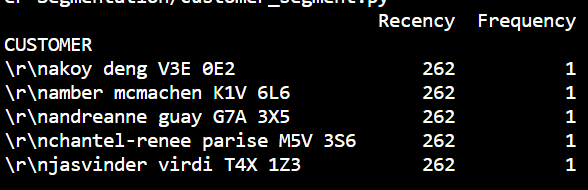
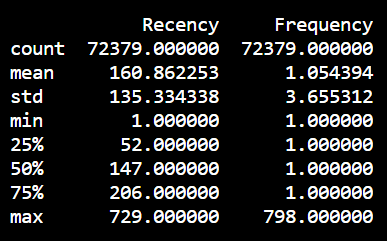
Created a column customer key ‘CUSTOMER’ by concatenating DESCRIPTION and ZIP CODE

# CALCULATION OF RFM METRICS

RFM Metrics are crucial for Customer Segmentation, and are defined as:

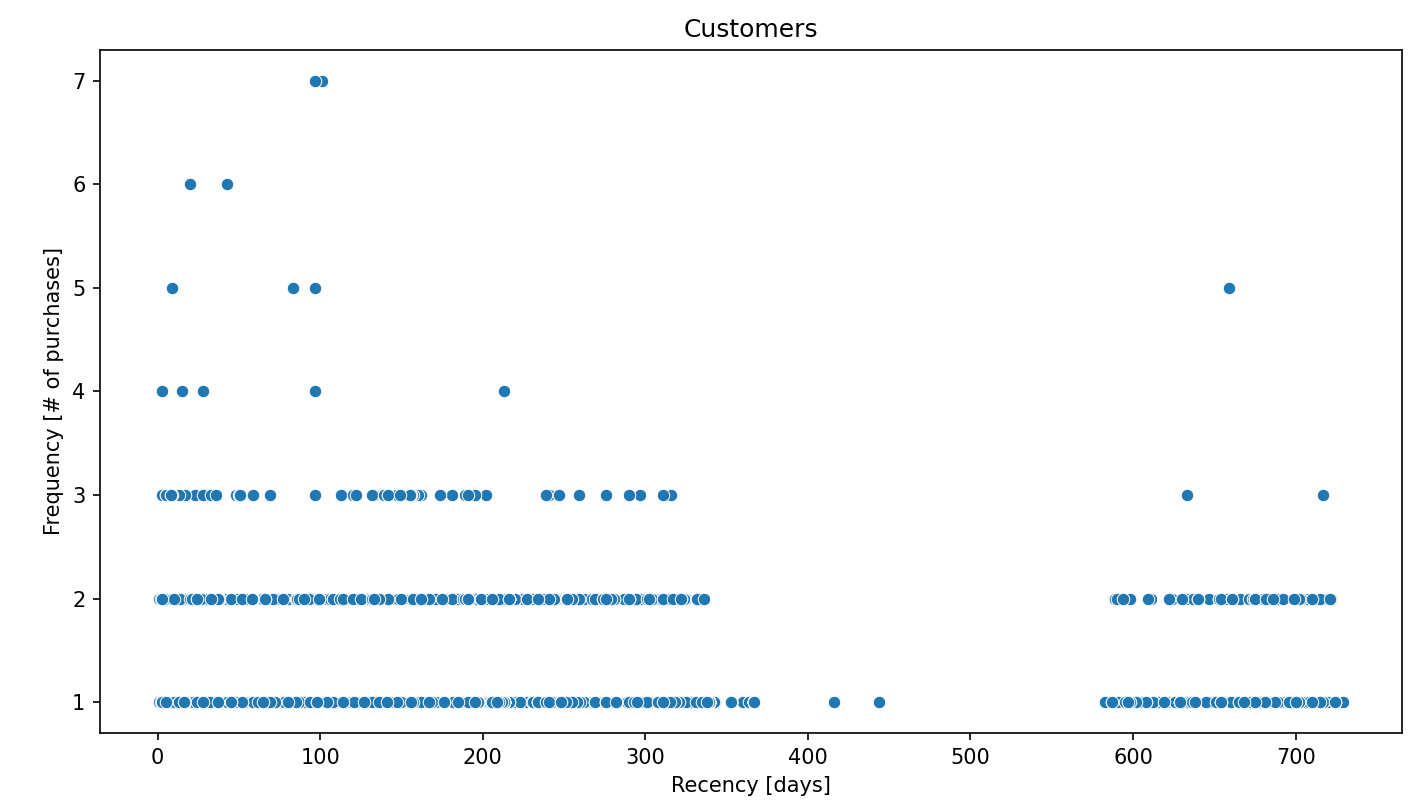
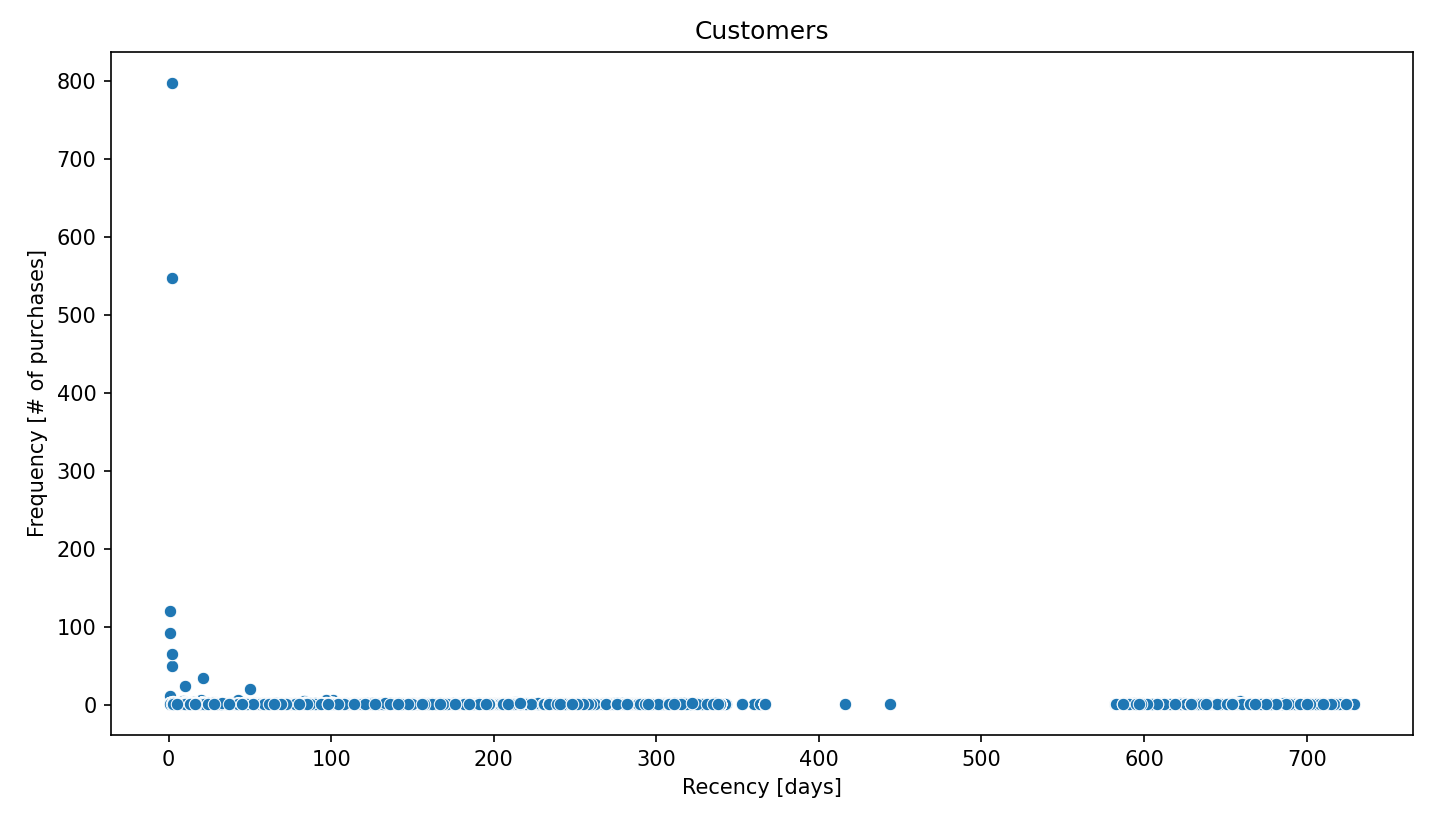
* **Recency:** days since the last purchase (considered today as the day after the most recent date in the data).
* **Frequency**: number of purchases during a specific time (in this case, the whole lifespan of the dataset is considered)
* **Monetary Value**: $ spent during a specific time **(not available)**

*Dataframe name: df\_rf* (72,379 x 2)

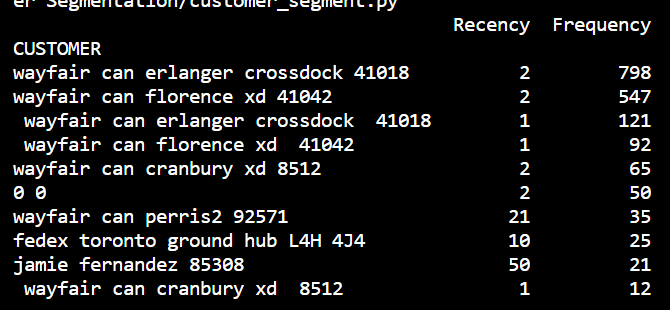
 

There are clearly potential outliers by the Frequency attribute. 2 Customers have more than 500 purchases and 8 more than 11.

We removed those customers with frequency more than 11 purchases. Only 11 outliers were removed (from ~72k)



High frequency customers are stored as they can be useful to the company. Only one of them has a natural name, the rest are companies.



Recommendations **(key ones):**

* **Remove all spaces before the first letter of each CUSTOMER row.**
* Add a separator like “-“ between customer and zip code

# PRE-PROCESSING THE DATA FOR K-MEANS CLUSTERING

## K-Means vs Pre-defined Segments

We will use K-Means Clustering instead of pre-defined segments for customer segmentation.

What is difference between segmentation and clustering? -> Both complement each other:

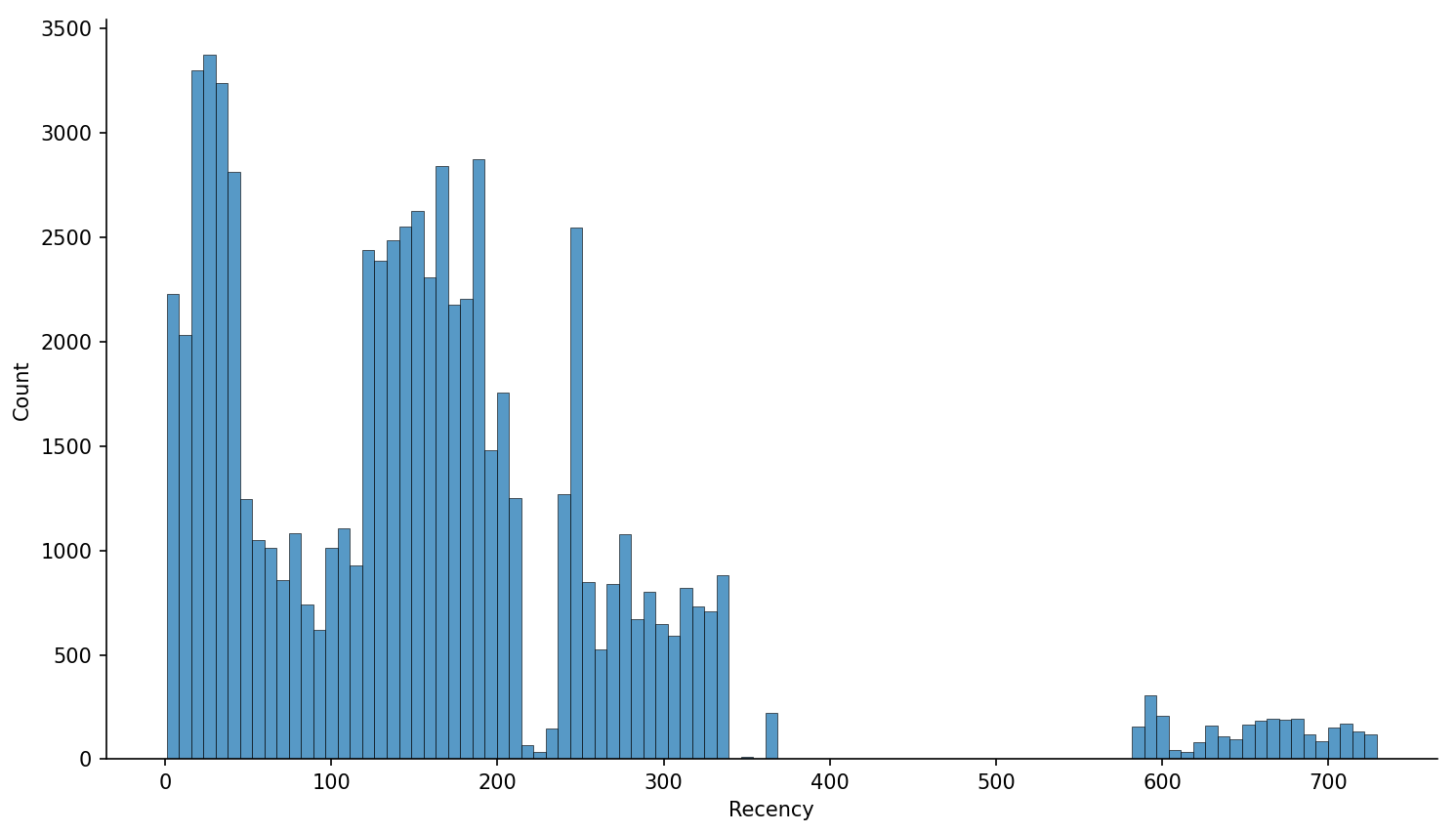
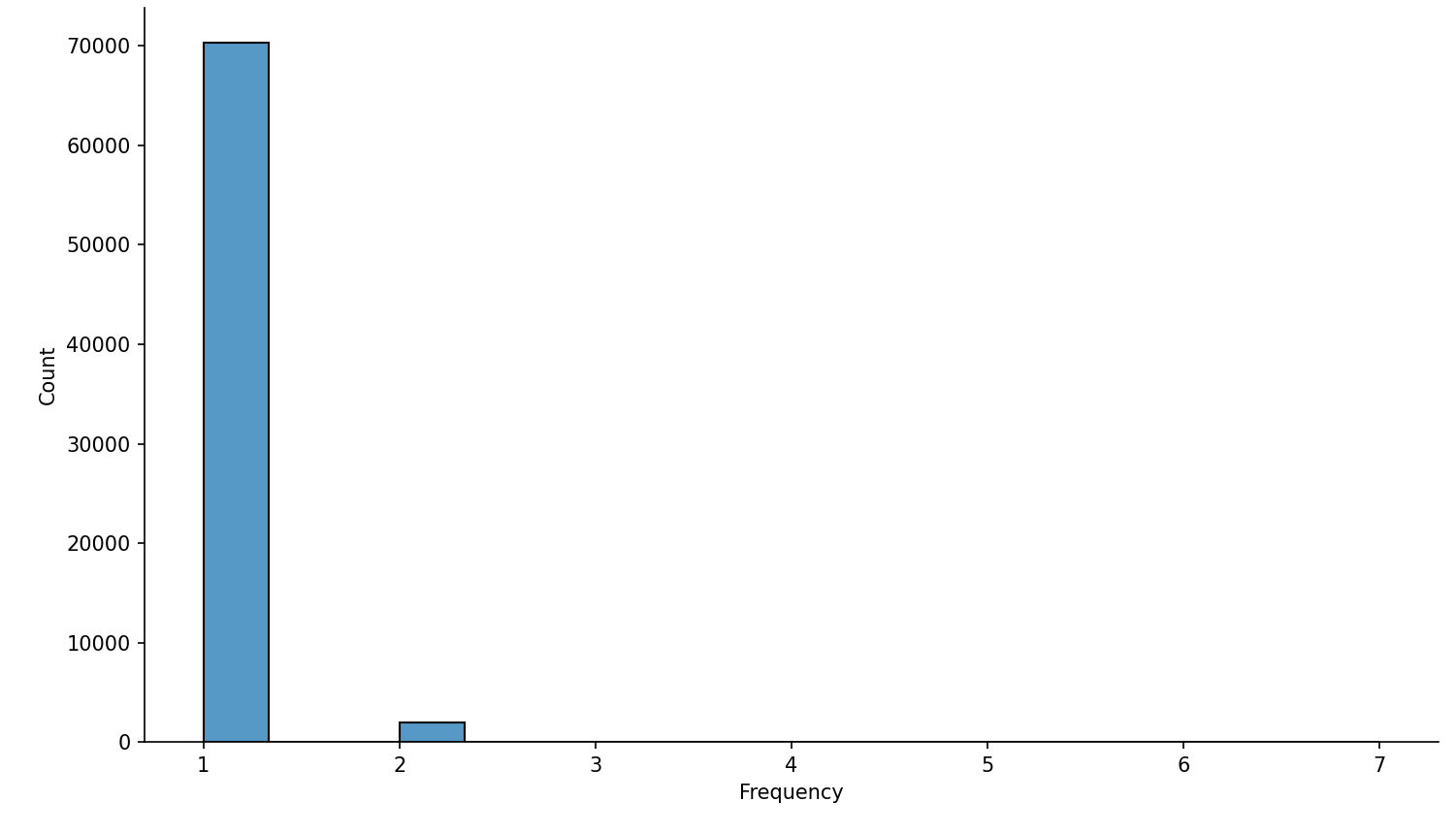
* Segmentation involves human-defined groupings
* Clustering involves ML-powered groupings

## Key K-Means Assumptions

* Symmetric distributions of variables
* Variables with same mean
* Variables with same variance

### Symmetric distributions of variables

Check skewness

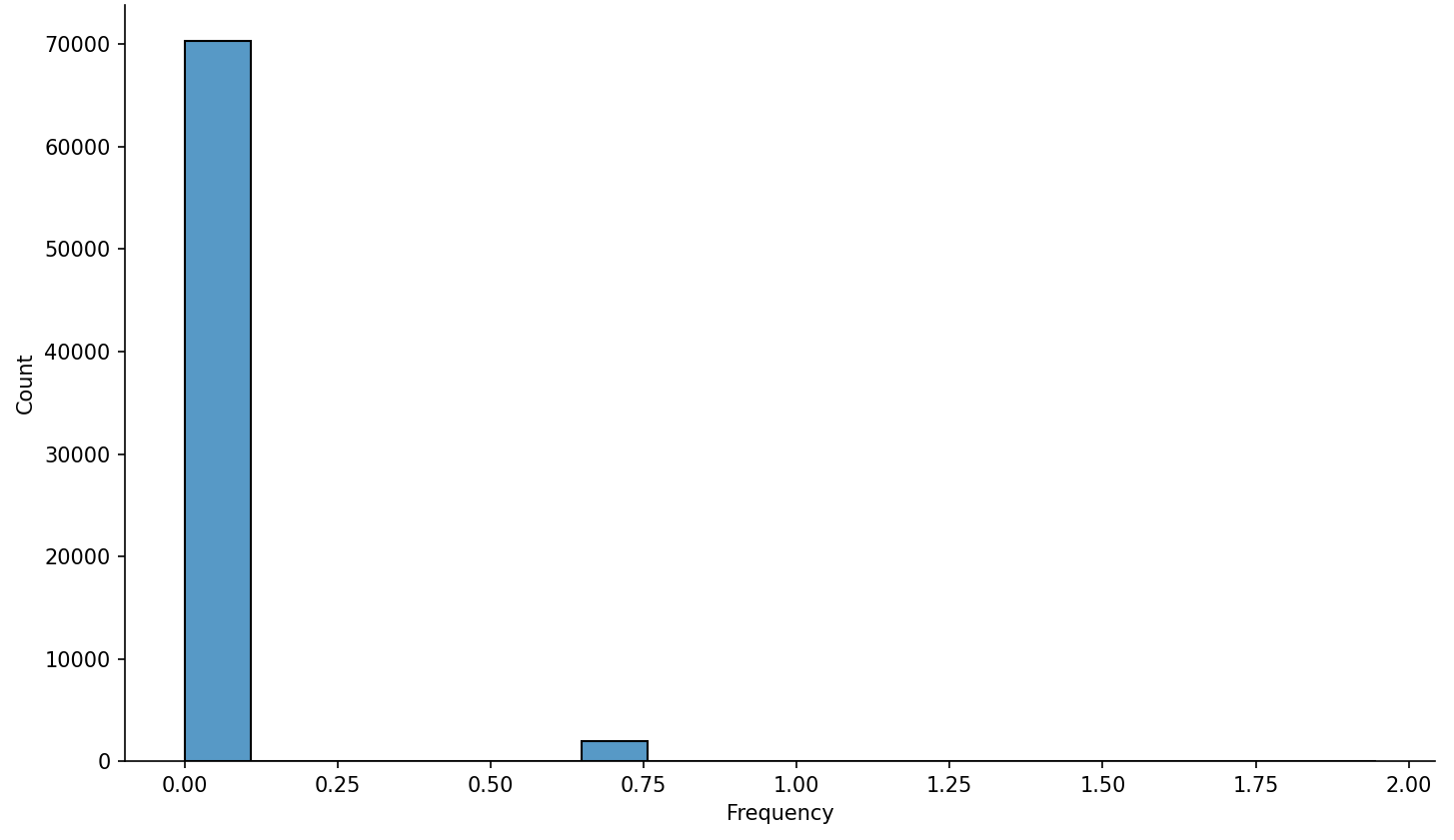
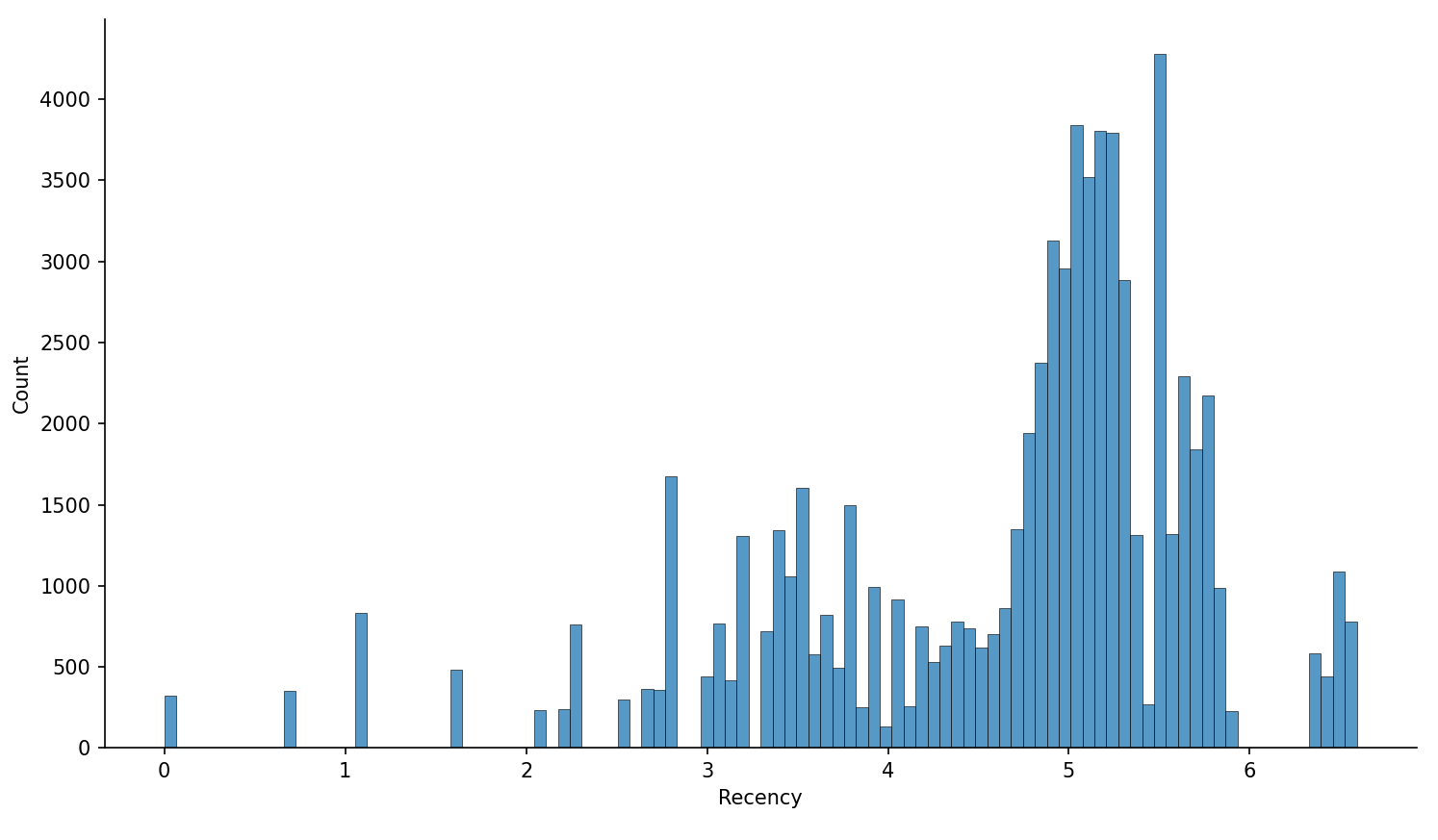
Both variables have a right skewed distribution

### Variables with same mean and variance

As we saw before, means are 160 and 1.05, and std 135 and 3.6, respectively for Recency and Frequency. So, we must normalize

**Unskew the data**

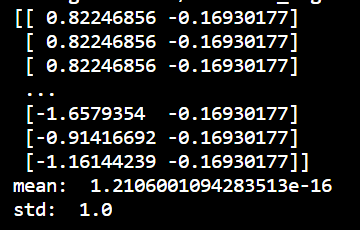
Applying log to the dataset, we get the following distributions:



We can see a distribution that is closer to symmetric for Recency, but no improvement for Frequency. For this reason, **final K-Means groups may be not as accurate as desired.**

**Normalize the variables**

We apply the StandardScaler() function to our log-transformed data, to get mean = 1 and std = 0 for both attributes:



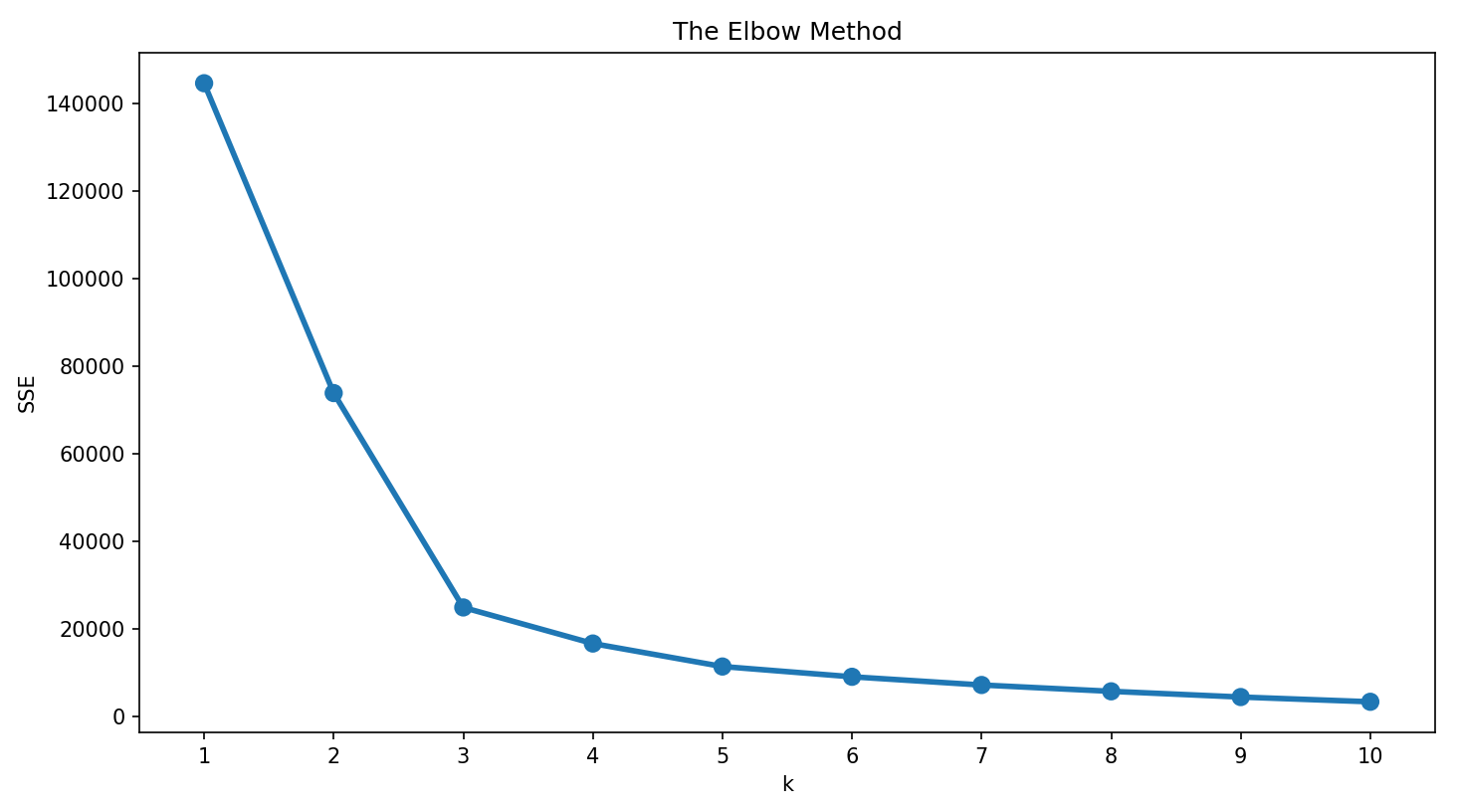
# RUNNING K-MEANS

## Choosing Number of Clusters: Elbow criterion

We ran the KMeans clustering simulation 10 times.

The following graph shows the Sum of Squared Errors (SSE) vs the number of clusters (k).

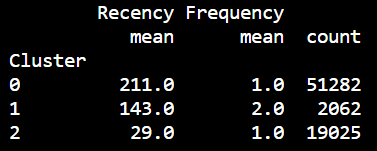
**The Elbow method suggest us to use 3 or 4 clusters. We will use 3.**



## Computing KMeans with 3 clusters

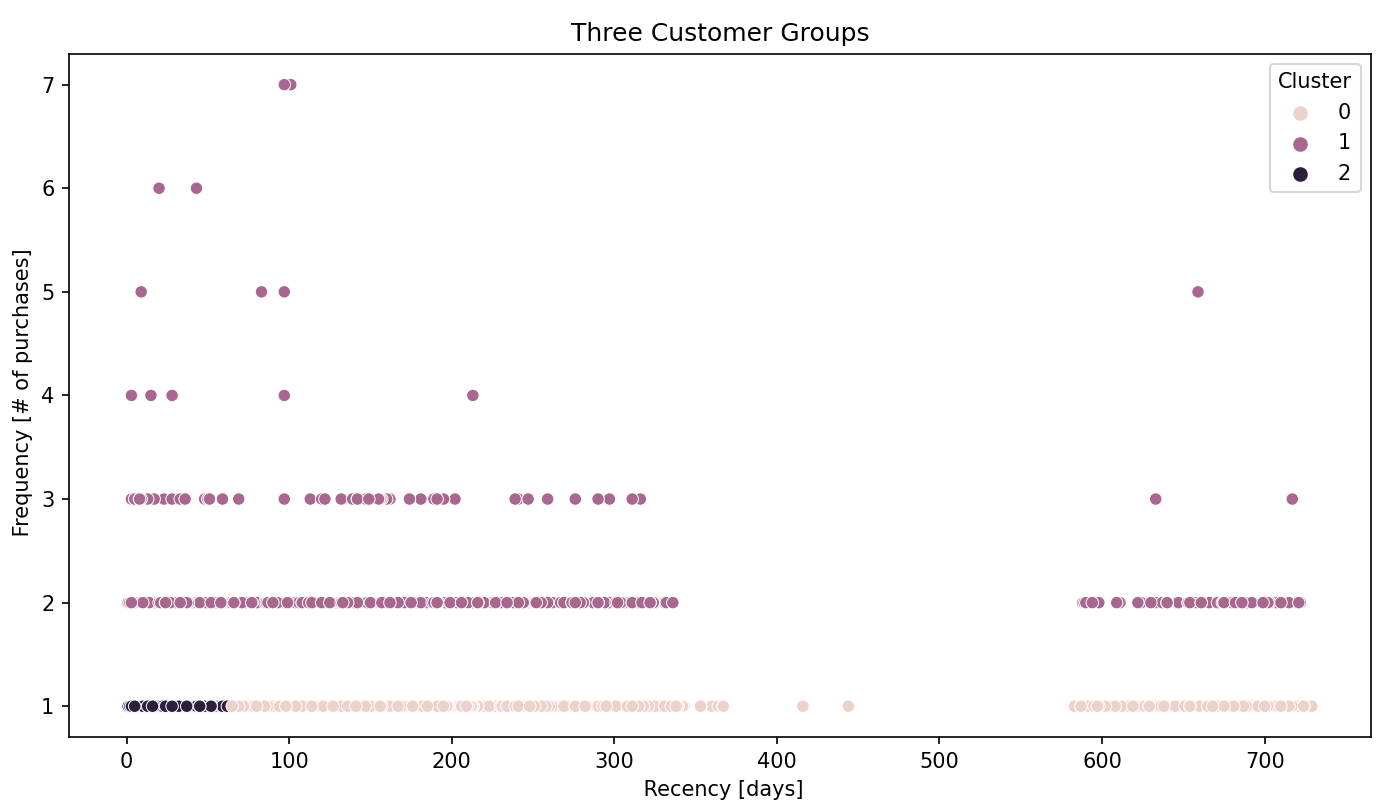
Running KMeans, assigning the cluster labels and computing summary statistics let us define and visualize 3 customer groups, based on Recency and Frequency:

*df\_rf\_k3\_stats:*



| **Customer Group** | **Group Size** | **Relevant Customer Attributes** |
| --- | --- | --- |
| 0 | 51k (71%) | Only 1 purchase and **longest** time without buying |
| 1 | 2k (3%) | More than 1 purchase |
| 2 | 19k (26%) | Only 1 purchase and **shortest** time without buying |

We can also visualize our customer groups in a 2D Chart, because we only have 2 Attributes. (N Attributes would require a N-dimensioned chart):

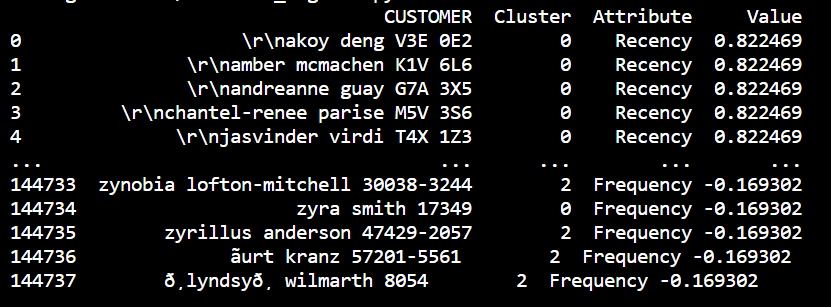


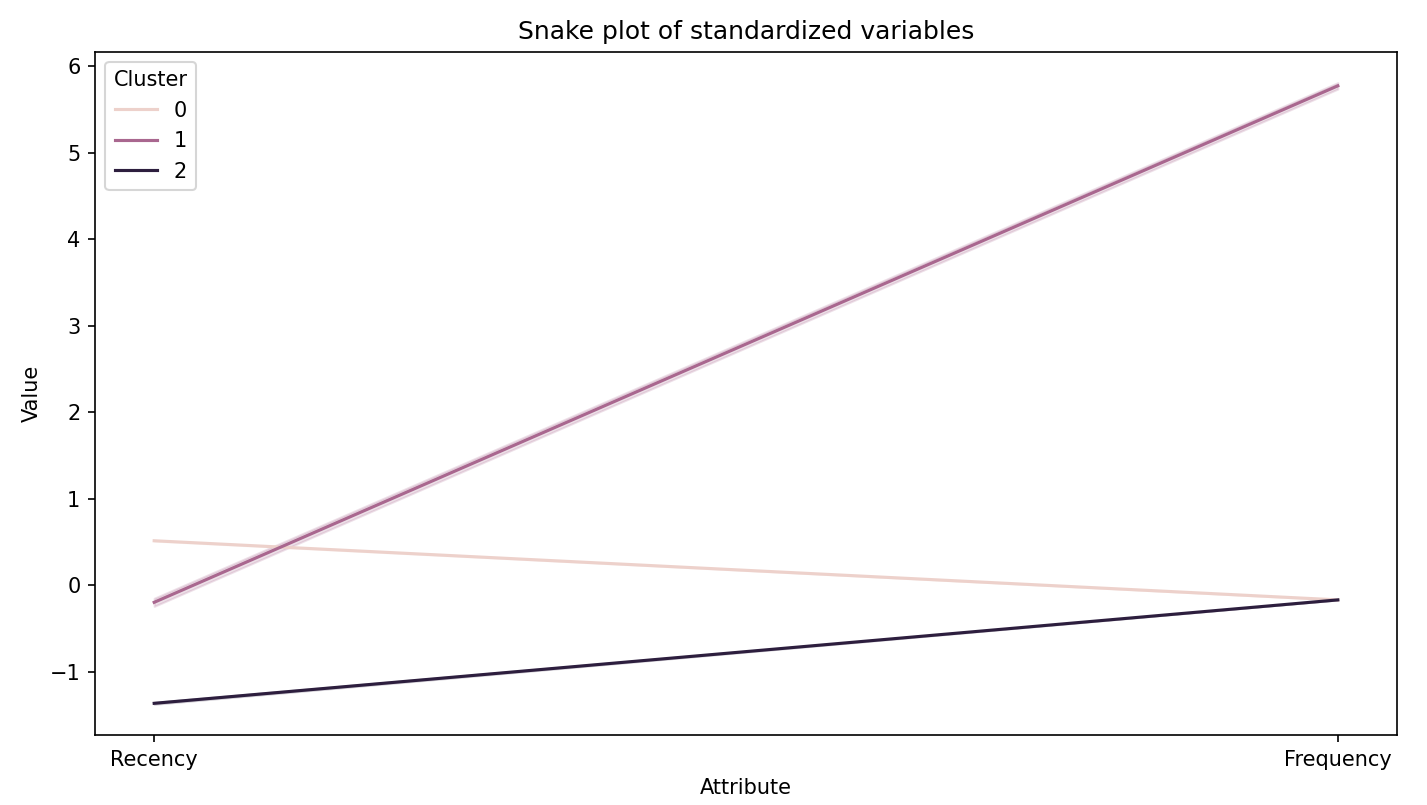
## Understand and compare groups

### Segment visualization

Snakeplots are charts that visualize RFM Values between the segments. It makes very easy and intuitive to interpret and compare the segments.

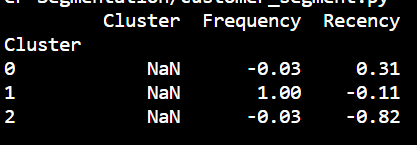
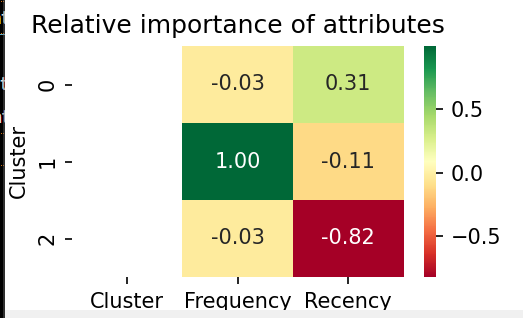
Data re-arengement to easily plot a snakeplot (*df\_melt):*





### Relative importance of attributes

Identify relative importance of each segment's attribute (using a heatmap):

As a ratio moves away from 0, attribute importance for a segment (relative to total population) increases.

* Frequency is by far the most important attribute for the Segment #1
* And recency for the Segment #2 and #0 (specially for segment #2)

# CONCLUSIONS AND RECOMMENDATIONS

* Get the price per purchase data so we can use **Monetary Value** and get more meaningful Customer segment description
* Only 2k customers (from ~70k) bought more than 1 time.
* Frequency shows a skewed distribution even after transformation with log, which implies that another transformation method should be used.
* Apply the recommendations mentioned in the calculation of RFM Metrics section of this doc.