# Μεθοδολογία

## Ερευνητική διαδικασία (θα συμπληρωθεί μετά την ολοκλήρωση της μεθόδου RFM)

## Το σύνολο δεδομένων: περιγραφή, σχολιασμός, και καθαρισμός (ολοκληρώθηκε 11/05/2024)

## Η μέθοδος RFM (είμαστε εδώ)

## Αποτελέσματα-Συζήτηση (επόμενη ενότητα προς ολοκλήρωση)

* Παράθεση και σύγκριση των RFM scores που βγήκαν από τις 2 μεθόδους
* 5^3 vs. 40 RFM scores

# 2. Το σύνολο δεδομένων

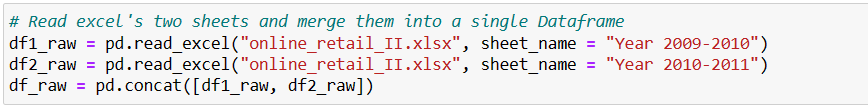
## 2.1 Περιγραφή συνόλου δεδομένων

Για την παρούσα ανάλυση χρησιμοποιήθηκαν δεδομένα από τις συναλλαγές πελατών ενός ηλεκτρονικού καταστήματος με είδη δώρων. Το κατάστημα εδρεύει στο Ηνωμένο Βασίλειο και εξυπηρετεί πελάτες λιανικής αλλά και χονδρικής. Το εν λόγω σύνολο δεδομένων αντλήθηκε από την ιστοσελίδα Kaggle[[1]](#footnote-1), ωστόσο είναι επίσης διαθέσιμο και στην ιστοσελίδα UCI Machine Learning Repository[[2]](#footnote-2), από όπου αντλούμε πληροφορίες για τις μεταβλητές του. Τα δεδομένα βρίσκονται σε μορφή excel και είναι χωρισμένα σε δύο καρτέλες (*Year 2009-2010*, *Year 2010-2011*)σύμφωνα με το έτος πραγματοποίησης των συναλλαγών*.* Οι συναλλαγές χαρακτηρίζονται από οκτώ μεταβλητές, οι οποίες παρουσιάζονται παρακάτω:

1. **Invoice**: Αριθμός τιμολογίου. Είναι εξαψήφιος για την κάθε συναλλαγή. Αν ξεκινάει με το γράμμα “c” υποδηλώνει ακύρωση τιμολογίου (cancellation).
2. **StockCode**: Κωδικός προϊόντος. Είναι ένας πενταψήφιος αριθμός, μοναδικός για κάθε προϊόν.
3. **Description**: Η ονομασία του προϊόντος.
4. **Quantity**: Η ποσότητα προϊόντων που αντιστοιχεί σε κάθε συναλλαγή.
5. **InvoiceDate**: Η ημερομηνία και η ώρα έκδοσης του τιμολογίου.
6. **Price**: Η τιμή μιας μονάδας προϊόντος εκφρασμένη σε λίρες (£).
7. **Customer ID**: Ο μοναδικός αριθμός καταχώρισης πελάτη. Αποτελείται από 5 ψηφία.
8. **Country**: Το όνομα της χώρας στην οποία κατοικεί ο κάθε πελάτης.

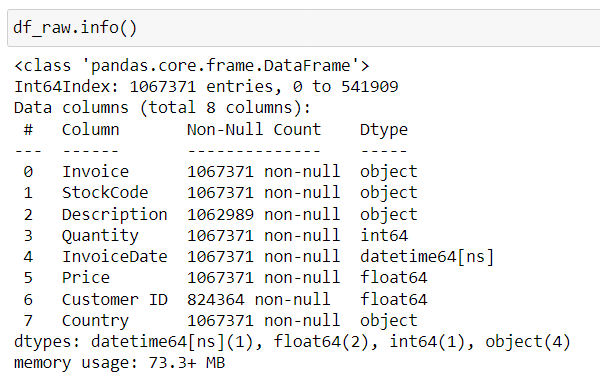
Για τον καθαρισμό του συνόλου δεδομένων, καθώς και για την πραγματοποίηση της περιγραφικής και διερευνητικής ανάλυσης των δεδομένων (EDA), χρησιμοποιήθηκε το Jupyter Notebook και το εργαλείο οπτικοποίησης δεδομένων Microsoft Power BI.

Αρχικά ενοποιήθηκαν τα δύο excel sheets με την χρήση των εντολών pd.read\_excel() και pd.concat() της Python σε ένα σύνολο δεδομένων, το οποίο ονομάστηκε df\_raw, όπου με pd συμβολίζεται η βιβλιοθήκη pandas της Python.



Ε?? Ενοποίηση των δύο υποσυνόλων δεδομένων με python

Παρακάτω παρατίθενται πληροφορίες για το σύνολο των εγγραφών (1 067 371) και των μεταβλητών (8) του ενοποιημένου dataset, όπως προκύπτει από την εντολή df\_raw.info() της Python:



Ε?? Πληροφορίες για τις μεταβλητές του ενοποιημένου συνόλου δεδομένων

Με την εντολή df\_raw.head(10) βλέπουμε ενδεικτικά τις δέκα πρώτες συναλλαγές του dataset:

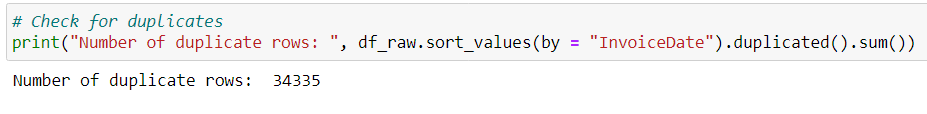


Ε?? Απεικόνιση των 10 πρώτων συναλλαγών του συνόλου δεδομένων

## 2.2 Διερευνητική ανάλυση δεδομένων και καθαρισμός δεδομένων

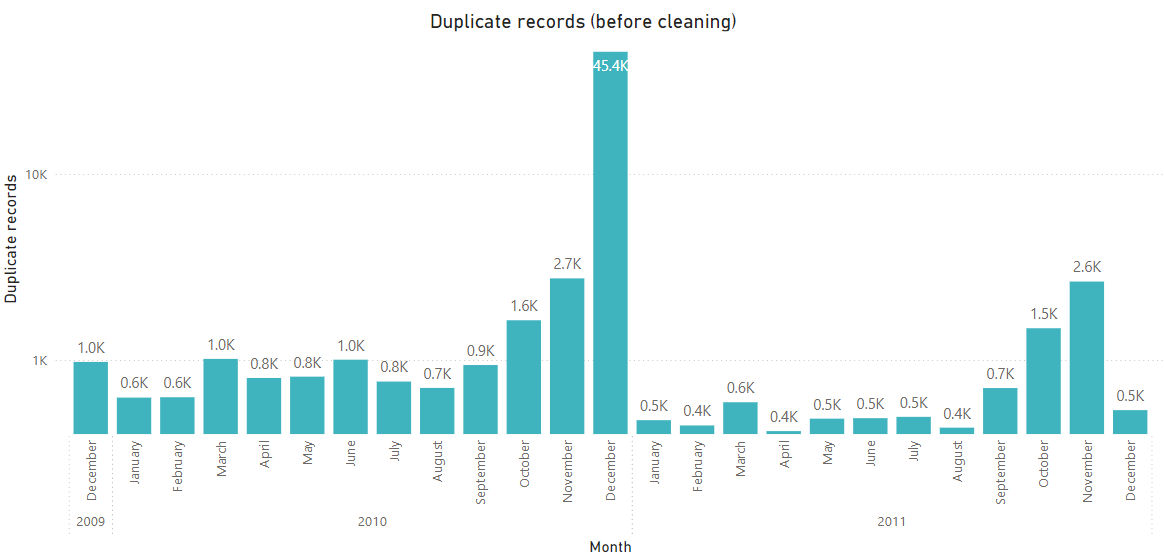
### 2.2.1 Αφαίρεση διπλών καταχωρίσεων

Το πρώτο βήμα είναι να ελεγχθεί αν υπάρχουν διπλές καταχωρίσεις. Με την εντολή df\_raw.sort\_values(by=“InvoiceDate”).duplicated().sum() παρατηρείται ότι ο αριθμός των διπλών καταχωρίσεων είναι 34 335.



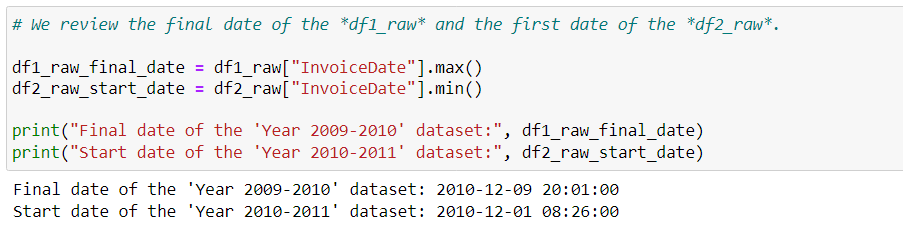
Ε?? Αριθμός διπλών καταχωρίσεων που προέκυψε από την ενοποίηση των δύο υποσυνόλων

Στο διάγραμμα ?? απεικονίζεται η κατανομή των διπλών καταχωρίσεων όπου διαπιστώνεται ότι το μεγαλύτερο μέρος αυτών παρατηρείται κατά το μήνα Δεκέμβριο 2010. Το γεγονός αυτό πιθανότατα προκλήθηκε κατά την ενοποίηση των δύο excel sheets, τα οποία περιείχαν κοινές συναλλαγές που αφορούσαν στον συγκεκριμένο μήνα.



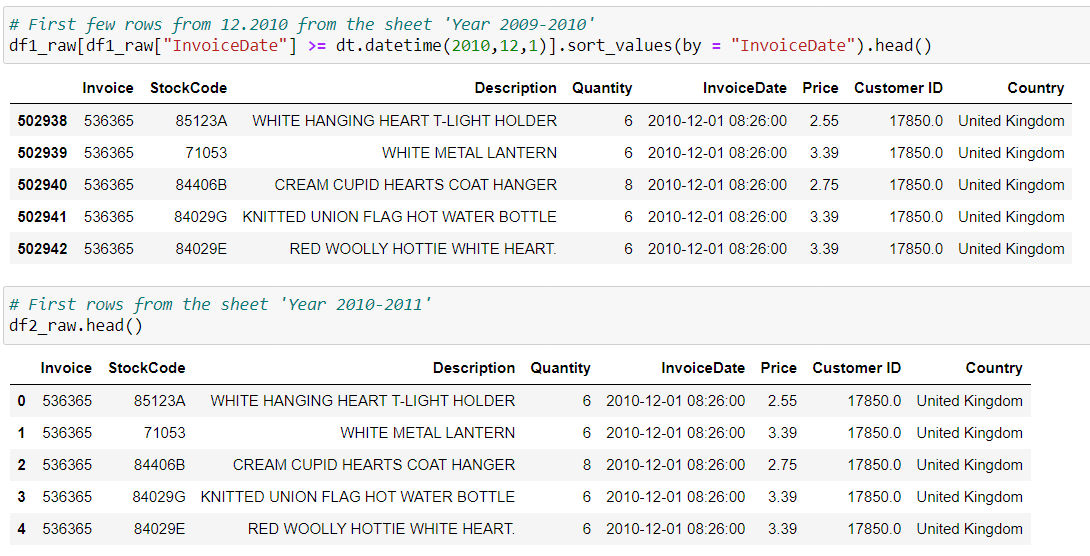
E?? Κατανομή διπλών καταχωρίσεων στους μήνες του συνόλου δεδομένων (πριν την αφαίρεση μέρους τους)

Για να εξακριβώσουμε την ισχύ της παραπάνω υπόθεσης, υπολογίζουμε την τελευταία ημερομηνία του df1\_raw (Year 2009-2010) και την πρώτη ημερομηνία του df2\_raw (Year 2010-2011). Όπως φαίνεται στην παρακάτω εικόνα, υπάρχουν εγγραφές που αφορούν το χρονικό διάστημα 01/12/2010 έως 09/12/2010 (9 ημέρες), οι οποίες συναντώνται και στους δύο πίνακες.



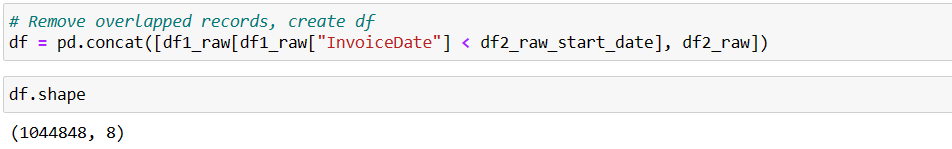
Ε?? Διερεύνηση των δύο υποσυνόλων δεδομένων για εντοπισμό overlap στις ημερομηνίες

Ταυτόχρονα μπορούμε να διαπιστώσουμε το παραπάνω γεγονός και οπτικά, εκτυπώνοντας από τον πίνακα df1\_raw τις συναλλαγές που πραγματοποιήθηκαν κατά το συγκεκριμένο χρονικό διάστημα, και από τον πίνακα df2\_raw τις πρώτες εγγραφές του.



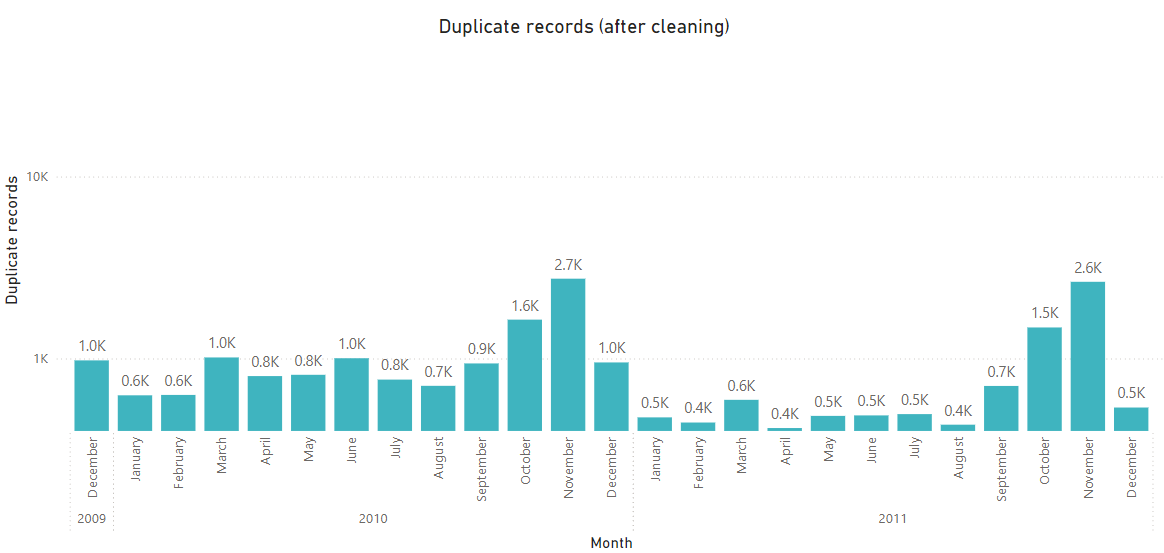
Ε?? Απεικόνιση των 5 πρώτων συναλλαγών που πραγματοποιήθηκαν κατά το διάστημα 01/12/2010 - 09/12/2010 στα δύο υποσύνολα δεδομένων

Μετά τις παραπάνω παρατηρήσεις, αφαιρούμε τις διπλές καταχωρίσεις μέσω της εντολής pd.concat(), η οποία έχει ως αποτέλεσμα τη δημιουργία του πίνακα **df**, ο οποίος περιέχει 1 044 848 συναλλαγές.



Ε?? Αφαίρεση των διπλών καταχωρίσεων που παρατηρήθηκαν τις πρώτες 9 ημέρες του Δεκεμβρίου 2010

Σε αυτό το σημείο ο αριθμός των διπλών καταχωρίσεων είναι 11 812, δηλαδή περίπου 66% λιγότερες σε σχέση με εκείνες του αρχικού dataset.

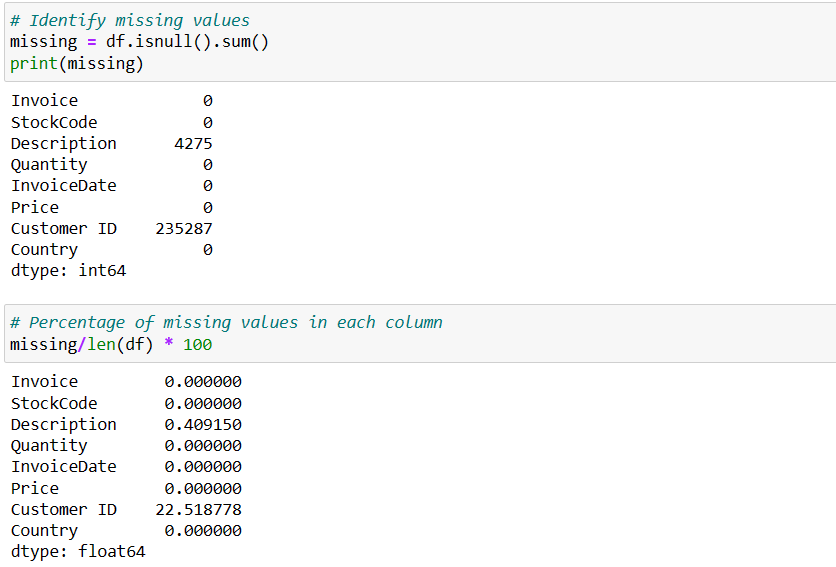


E?? Κατανομή διπλών καταχωρίσεων στους μήνες του συνόλου δεδομένων (μετά την αφαίρεση μέρους τους)

Σε πραγματικές συνθήκες θα ζητούσαμε από τους υπεύθυνους συλλογής των δεδομένων του dataset να μας εξηγήσουν τί σημαίνουν οι διπλές αυτές καταχωρίσεις. Συγκεκριμένα, δεδομένου ότι η ποσότητα του κάθε προϊόντος σε κάθε μία από αυτές τις διπλές καταχωρίσεις είναι 1, μπορούμε να σχηματίσουμε την υπόθεση ότι για αυτές τις συναλλαγές το σύστημα καταχωρίσεων της επιχείρησης αντί να αθροίσει τις ποσότητες του κάθε προϊόντος για κάθε παραγγελία, τις καταχώρισε ξεχωριστά σε ισόποσες εγγραφές. Εφόσον δεν υπάρχει κάποιο ανασταλτικό επιχείρημα που να καταρρίπτει αυτή την υπόθεση, θα συμπεριλάβουμε το υπόλοιπο 34% των διπλών καταχωρίσεων στην ανάλυση που ακολουθεί.

### 2.2.2 Ελλείπουσες τιμές

Χρησιμοποιώντας την εντολή df.isnull().sum() της Python παρατηρούμε τον αριθμό των ελλειπουσών τιμών για κάθε μεταβλητή. Οι μόνες μεταβλητές που έχουν missing values είναι η μεταβλητή Description (4 275) και η μεταβλητή Customer ID (235 287). Για να αποκτήσουμε εικόνα του σχετικού μεγέθους των ελλειπουσών τιμών της κάθε μεταβλητής χρησιμοποιούμε την εντολή missing/len(df) \* 100, όπου missing είναι το αποτέλεσμα της προηγούμενης εντολής που εκτελέσαμε.

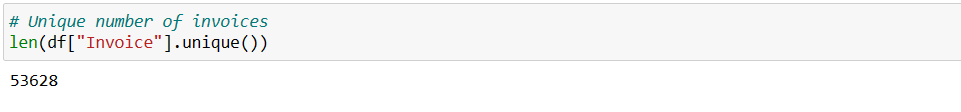


Ε?? Προσδιορισμός ελλειπουσών τιμών ανά μεταβλητή

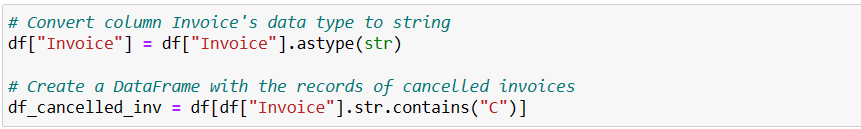
Προτού ξεκινήσουμε την διαμόρφωση του συνόλου δεδομένων για τις ανάγκες της παρούσας ανάλυσης θα πρέπει να διερευνηθούν οι μεταβλητές του και να αντληθούν χρήσιμες πληροφορίες. Στην επόμενη ενότητα ακολουθεί η διερευνητική ανάλυση του συνόλου δεδομένων.

### 2.2.3 Ανάλυση των μεταβλητών του συνόλου δεδομένων

#### *2.2.3.1 Στήλη Invoice*

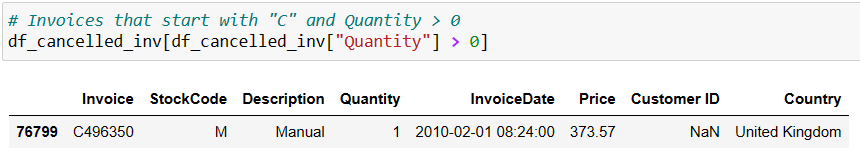
Ο μοναδικός αριθμός των τιμολογίων είναι 53 628.

Μέσω της περιγραφής που έχει δώσει ο ίδιος ο εκδότης του dataset πληροφορούμαστε ότι ορισμένα από τα τιμολόγια αφορούν ακυρώσεις, και συγκεκριμένα όσα ξεκινούν με το γράμμα “C”. Θα επικεντρωθούμε αρχικά στις ακυρώσεις, και έπειτα στο πώς αυτές συνδέονται με τις μεταβλητές Quantity και Price. Για τον σκοπό αυτό δημιουργήσαμε ένα DataFrame[[3]](#footnote-3) με τις συναλλαγές που αφορούν ακυρώσεις τιμολογίων όπως φαίνεται στην εικόνα ??. Ο αριθμός των ακυρώσεων είναι 19 165 και αντιστοιχεί στο 1,8% του συνολικού αριθμού των συναλλαγών του dataset.



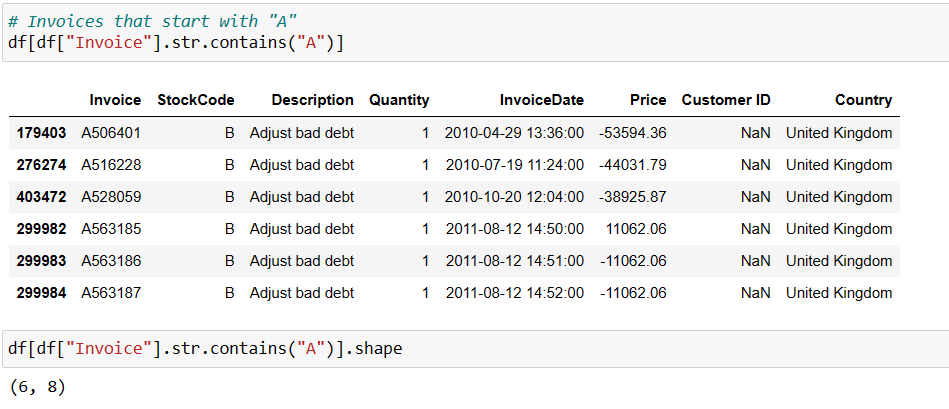
Ε?? Συναλλαγές που αφορούν ακυρώσεις τιμολογίων

Από τις ακυρώσεις τιμολογίων μόνο μία συναλλαγή παρουσιάζει θετική τιμή στη στήλη Quantity, ενώ οι υπόλοιπες έχουν αρνητική τιμή. Η εν λόγω συναλλαγή έχει ελλείπουσα τιμή στη στήλη Customer ID, όπως φαίνεται και στην εικόνα ??.



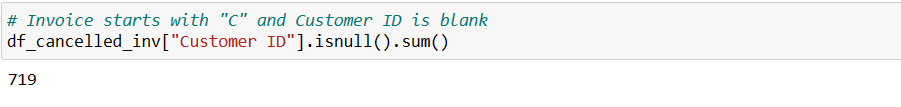
Ε?? Συναλλαγές με ακυρωτικά τιμολόγια και θετική τιμή στη στήλη Quantity

Εκτός από τα τιμολόγια που ξεκινούν από “C” υπάρχουν και κάποια που ξεκινούν από “A”. Τα τιμολόγια αυτά είναι μόνο 6 και έχουν missing value στη στήλη Customer ID. Τα 5 από αυτά έχουν αρνητική τιμή στην στήλη Price και μοναδιαία τιμή στην στήλη Quantity. Παρατηρώντας το περιεχόμενο της στήλης Description για κάθε ένα από αυτά τα 6 τιμολόγια (“Adjust bad debt”) ερμηνεύουμε πως πρόκειται για ενέργειες λογιστικής φύσεως και συνεπώς ότι δεν σχετίζονται με τις πωλήσεις του ηλεκτρονικού καταστήματος. Επιπλέον παρατηρούμε ότι υπάρχει μία συσχέτιση μεταξύ των τιμολογίων “A563185”, “A563186”, και “A563187”. Στην εικόνα ?? παρατίθενται λεπτομέρειες για τα εν λόγω 6 τιμολόγια.



E?? Απεικόνιση εγγραφών με τιμολόγια που ξεκινούν από "A"

Από τα 19 165 ακυρωμένα τιμολόγια τα 719 (3%) έχουν missing value στην στήλη Customer ID.



E?? Ακυρωμένα τιμολόγια με ελλείπουσα τιμή στη στήλη CustomerID

#### *2.2.3.2 Στήλη Quantity*

Η στήλη Quantity αναφέρεται είτε στην ποσότητα των προϊόντων που αγοράστηκαν σε κάθε συναλλαγή (Quantity > 0) είτε, όπως είδαμε και στην ανάλυση της στήλης Invoice, σε επιστροφές προϊόντων (Quantity < 0). Παρατηρούμε ότι δεν υπάρχουν ελλείπουσες τιμές στην στήλη Quantity και συνεπώς δεν χρειάζεται να προβούμε σε κάποια διαγραφή για την ανάλυση που θα ακολουθήσει.

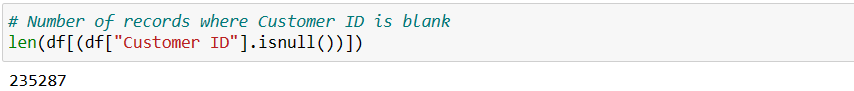


Ε?? Αριθμός εγγραφών με αρνητική τιμή στη στήλη Quantity

Οι εγγραφές με αρνητική τιμή στην στήλη Quantity είναι συνολικά 22 557. Ο αριθμός αυτός ξεπερνάει τις εγγραφές των τιμολογίων που ξεκινούν από “C” και έχουν αρνητική τιμή (19164). Αυτό υποδηλώνει ότι η αρνητική τιμή στην ποσότητα δεν σημαίνει απαραίτητα μόνο την επιστροφή κάποιου προϊόντος, αλλά ενδέχεται να είναι συνδεδεμένη και με άλλα είδη συναλλαγών, όπως ακυρώσεις πωλήσεων ή ελαττωματικά προϊόντα.

#### *2.2.3.3 Στήλη Customer ID*

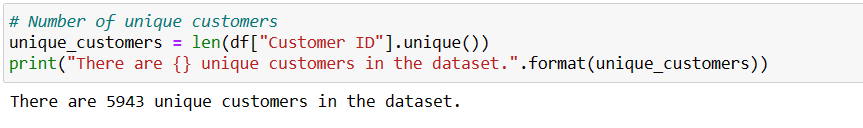
H στήλη Customer ID αναφέρεται στον μοναδικό αριθμό πελάτη (αναγνωριστικό πελάτη). Η στήλη αυτή περιέχει 235 287 missing values, δηλαδή περίπου το 22,5% των εγγραφών. Το ποσοστό είναι σημαντικά μεγάλο για να αγνοηθεί, και ανάλογα με την ανάλυση που θα πραγματοποιηθεί στα δεδομένα απαιτείται και διαφορετικός χειρισμός των missing values.



E?? Αριθμός εγγραφών με ελλείπουσα τιμή στη στήλη CustomerID

Σε πραγματικές συνθήκες θα ζητούσαμε από τους υπεύθυνους συλλογής των δεδομένων του dataset να μας εξηγήσουν γιατί οι συγκεκριμένες εγγραφές δεν έχουν Customer ID. Αφού κάτι τέτοιο δεν είναι εφικτό, μπορούμε να υποθέσουμε ότι οι συγκεκριμένοι πελάτες δεν πραγματοποίησαν ποτέ εγγραφή στο online κατάστημα και επομένως δεν τους αποδόθηκε αναγνωριστικό πελάτη.

Ο αριθμός των μοναδικών Customer ID του dataset που εξετάζουμε είναι5 943, στον οποίο αριθμό φτάνουμε με χρήση της εντολής της Python που παρουσιάζεται στην εικόνα ??.



E?? Αριθμός μοναδικών τιμών στη στήλη CustomerID

Ο αριθμός αυτός αντιπροσωπεύει 5 942 εγγεγραμμένους πελάτες συν όλους εκείνους οι οποίοι δεν πραγματοποίησαν ποτέ εγγραφή και των οποίων το Customer ID συμβολίζεται με τον κενό χαρακτήρα.

Οι ελλείπουσες τιμές της στήλης Customer ID θα αφαιρεθούν στην επόμενη ενότητα για τους σκοπούς της ανάλυσης RFM με στόχο την τμηματοποίηση των πελατών.

#### *2.2.3.4 Στήλη Country*

A screenshot of a computer

Description automatically generatedΟι συναλλαγές του dataset προέρχονται από 42 χώρες. Ενώ δεν παρατηρούνται missing values στην στήλη Country, υπάρχουν 756 εγγραφές στις οποίες η χώρα είναι “Unspecified”.

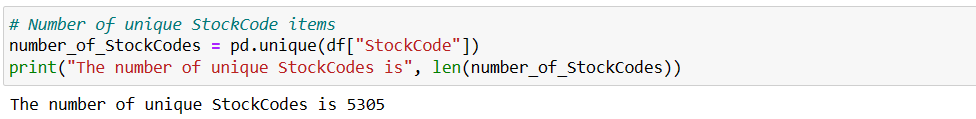
A graph with numbers and a bar

Description automatically generatedΤο μεγαλύτερο μερίδιο των συναλλαγών πραγματοποιήθηκε στο Ηνωμένο Βασίλειο (91.87%), ύστερα στην Ιρλανδία (1.69%) και στη Γερμανία (1.66%). Στο παρακάτω γράφημα παρουσιάζονται οι 10 χώρες με τον μεγαλύτερο αριθμό συναλλαγών.

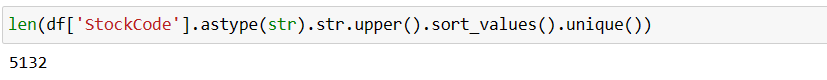
#### *2.2.3.5 Στήλη StockCode*

Η στήλη StockCode αναφέρεται στους κωδικούς προϊόντων που αγοράστηκαν. Η στήλη StockCode δεν περιέχει ελλείπουσες τιμές.

Κάτι το οποίο θα είχε αξία να ερευνηθεί είναι ποιοι κωδικοί έχουν τις περισσότερες πωλήσεις κατά το χρονικό διάστημα που εξετάζεται. Προς αυτόν τον σκοπό εκτελέστηκε η εντολή df.unique() της Python, σύμφωνα με την οποία οι μοναδικές τιμές της στήλης StockCode είναι 5 305.



Παράλληλα εκτελέσαμε την αντίστοιχη εντολή στο Power BI χρησιμοποιώντας την έκφραση DISTINCTCOUNTNOBLANKS της γλώσσας DAX, όπου διαπιστώσαμε ότι ο μοναδικός αριθμός προϊόντων διαφέρει (5 131 σε σχέση με 5 305). Είναι κρίσιμο να μάθουμε γιατί είναι διαφορετικά τα αποτελέσματα που προκύπτουν από τα δύο εργαλεία, και να φέρουμε τις τιμές σε συμφωνία, γιατί αλλιώς οι επερχόμενες αναλύσεις μέσω Python και Power BI θα αποκλίνουν και θα αποφέρουν πιθανώς διαφορετικά συμπεράσματα και συνεπώς αμφισημία.

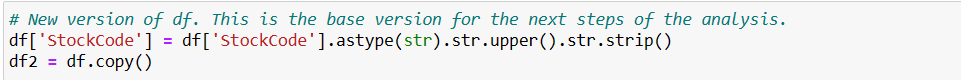
Δεδομένου ότι η Python αναφέρει περισσότερους μοναδικούς κωδικούς είναι λογικό να υποθέσουμε ότι η Python διάβασε κάποιες τιμές της στήλης ως μοναδικές, κάτι που δεν έκανε το Power BI, διαφορετικά δεν θα υπήρχε αυτή η ασυμφωνία. Με σκοπό να ερευνήσουμε το αν είναι σωστή αυτή η υποψία πρέπει αρχικά να φέρουμε τις τιμές της στήλης σε ένα κοινό επίπεδο ώστε να είναι συγκρίσιμες μεταξύ τους. Συγκεκριμένα, μέσω Python μετατρέπουμε το data type όλων των τιμών της στήλης StockCode σε τύπο string και έπειτα όλους τους χαρακτήρες των strings σε κεφαλαία γράμματα: με αυτό τον τρόπο όλες οι τιμές μπορούν να συγκριθούν μεταξύ τους. Σε αυτό το σημείο μετράμε εκ νέου τις μοναδικές τιμές της στήλης και το αποτέλεσμα που προκύπτει είναι 5 132, δηλαδή μία παραπάνω από αυτές του Power BI.

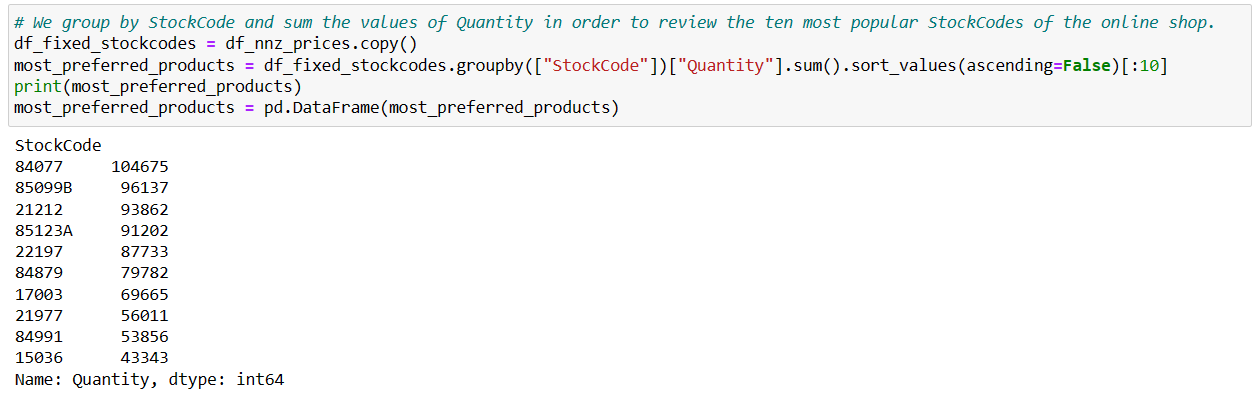
Για να εντοπίσουμε τη μία τιμή που εμφανίζεται ως διαφορά, παραθέσαμε τη λίστα των μοναδικών τιμών που προέκυψαν απότην Python και από το PowerBI σε δύο διαφορετικά αρχεία. Στη συνέχεια ταξινομήσαμε αλφαβητικά τις τιμές που περιέχονται στα δύο αρχεία και αντιπαραβάλλοντάς τες εντοπίσαμε ότι η εγγραφή "47503J" εμφανιζόταν με δύο τρόπους στα αποτελέσματα της Python: "47503J " και "47503J" (στην πρώτη περίπτωση μετά τον χαρακτήρα J υπάρχει ο κενός χαρακτήρας). Στη συνέχεια χρησιμοποιήσαμε την εντολή str.strip() για να αφαιρέσουμε το κενό που δημιουργήθηκε στην ονομασία και στο τέλος επιβεβαιώσαμε ότι οι μοναδικές τιμές της στήλης StockCode είναι 5 131 και για τα δύο εργαλεία.

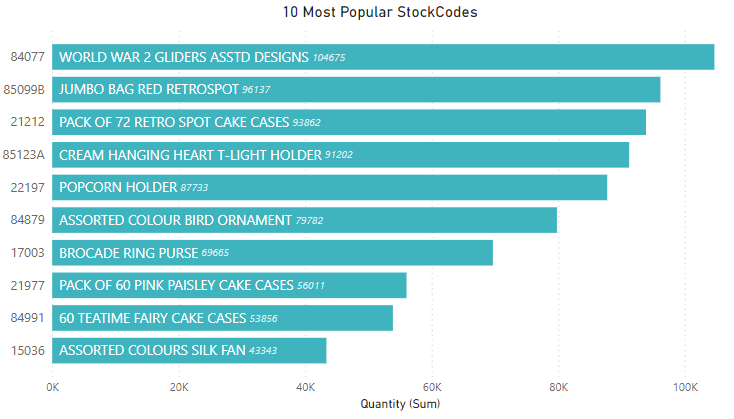
A screenshot of a computer code

Description automatically generated

Σημειώνουμε ότι δεν χρειάστηκε να κάνουμε όλα τα παραπάνω βήματα στο Power BI για να καταλήξουμε στον ίδιο αριθμό, καθώς η εντολή DISTINCTCOUNTNOBLANKS έκανε έναν αριθμό από υποθέσεις, οικεία βουλήσει, αγνοώντας για παράδειγμα τη διαφορά ανάμεσα στα πεζά και τα κεφαλαία γράμματα ή τους προπορευόμενους και λήγοντες κενούς χαρακτήρες στις ονομασίες των StockCode. Αν και στην συγκεκριμένη περίπτωση μπορεί να θεωρηθεί πλεονέκτημα, δεν σημαίνει ότι σε οποιαδήποτε ανάλυση θα καταλήγαμε στο ίδιο συμπέρασμα. Στο συγκεκριμένο dataset έχουμε να κάνουμε με μοναδικούς αριθμούς προϊόντων που είναι ξεκάθαροι και πεπερασμένοι, ενώ σε κάποιο άλλο dataset μια μικρή διαφορά στην ονομασία των προϊόντων θα μπορούσε να υποδηλώνει δύοαντικείμενα διακριτά. Κάθε dataset είναι ξεχωριστό και η εις βάθος κατανόηση των στοιχείων του είναι κρίσιμη προτού ο αναλυτής προβεί στον χειρισμό του.

Στην συνέχεια δημιουργήθηκε ένα νέο DataFrame, το df2, το οποίο περιλαμβάνει τις νέες τιμές της στήλης StockCode έπειτα από τις τροποποιήσεις που εφαρμόστηκαν πιο πάνω.

Κρίνουμε πως ο υπολογισμός των 10 πιο δημοφιλών προϊόντων του dataset θα είχε αξία για τον ιδιοκτήτη του καταστήματος και συνεπώς στην συνέχεια προβαίνουμε ακριβώς σε αυτή την ανάλυση. Αρχικά πρέπει να αποκλείσουμε τα προϊόντα που έχουν μηδενική τιμή στην στήλη Price, καθώς τα συγκεκριμένα είναι προϊόντα που δεν πωλήθηκαν αλλά προσφέρθηκαν ως δώρα ή προσφορές από το ηλεκτρονικό κατάστημα, και συνεπώς δεν υποδηλώνουν τις προτιμήσεις των πελατών. Το ίδιο θα κάναμε και στην περίπτωση που είχαμε ελλείπουσες τιμές στη στήλη Price, κάτι το οποίο δεν συμβαίνει στο παρόν dataset. Στην επόμενη εικόνα η συλλογή των προϊόντων με μη μηδενική τιμή συμβολίζεται με το DataFrame df\_nnz\_prices. Εκτελώντας τις υπόλοιπες εντολές της εικόνας φτάνουμε στη λίστα με τα δέκα πιο ευπώλητα προϊόντα, η οποία περιλαμβάνεται και αυτή στην ίδια εικόνα. Ακολουθώντας την ίδια ανάλυση μέσω Power BI καταλήγουμε στο ραβδόγραμμα της μεθεπόμενης εικόνας. Παρατηρούμε ότι οι δύο ξεχωριστές αναλύσεις συμπίπτουν.

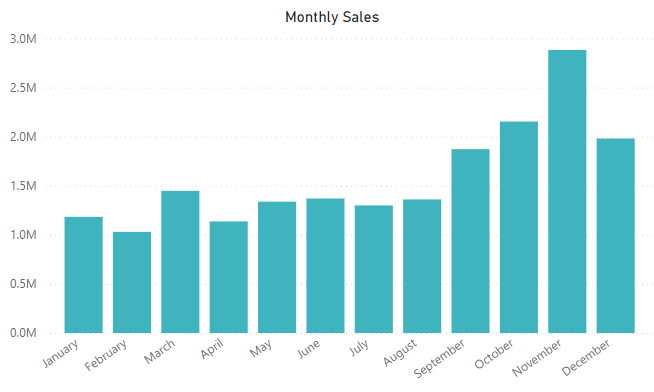


#### *2.2.3.6 Στήλη InvoiceDate*

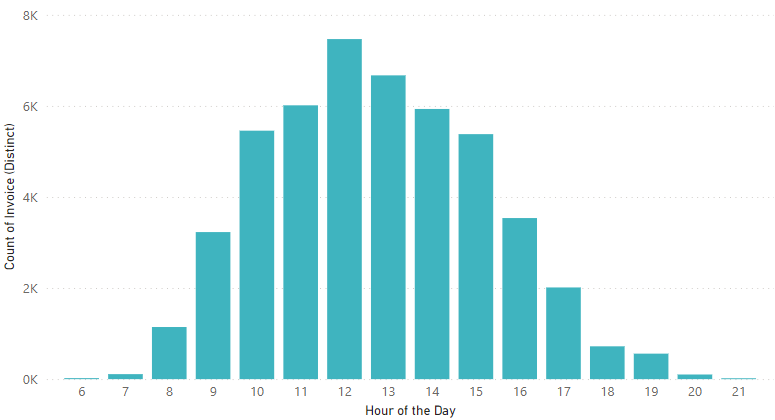
Η στήλη InvoiceDate αναφέρεται στην ημερομηνία και ώρα έκδοσης κάθε τιμολογίου. Όπως αναφέρθηκε και στην αρχική περιγραφή του dataset οι συναλλαγές εκτείνονται από 01/12/2009 έως 09/12/2011. Με σκοπό να ερευνηθεί ποιοι μήνες είχαν τις περισσότερες πωλήσεις δημιουργήθηκαν δύο νέες στήλες: η στήλη Month και η στήλη TotalAmount, οι οποίες προσαρτήθηκαν στο dataset. Η στήλη **Month** δημιουργήθηκε με την χρήση της βιβλιοθήκης datetime της Python, ενώ η στήλη **TotalAmount** προέκυψε από το γινόμενο των στηλών Price και Quantity.

A computer screen shot of text

Description automatically generated

Στο παρακάτω ραβδόγραμμα απεικονίζονται οι πωλήσεις ανά μήνα. Παρατηρήθηκε ότι οι μήνες Οκτώβριος, Νοέμβριος και Δεκέμβριος ήταν οι μήνες με τις περισσότερες πωλήσεις.

Επιπλέον δημιουργήθηκε η στήλη Hourώστενα ερευνηθεί ποιες ώρες κατά την διάρκεια της ημέρας έγιναν οι περισσότερες παραγγελίες. Σύμφωνα με το γράφημα που ακολουθεί ο μεγαλύτερος όγκος των παραγγελιών πραγματοποιείται ανάμεσα στις 10 π.μ. και στις 3 μ.μ.



Για τους σκοπούς της ανάλυσης που θα ακολουθήσει θα αφαιρέσουμε τα στοιχεία της ώρας από την στήλη InvoiceDate, της οποίας οι τιμές θα εμφανίζονται στο εξής στη μορφή «έτος-μήνας-ημέρα».

#### *2.2.3.7 Συνδυασμοί Customer ID και StockCode*

Όπως αναφέρθηκε στην περιγραφή του dataset, συχνά παρατηρείται ότιο ίδιος πελάτης αγόρασε ένα προϊόν σε διαφορετικές ποσότητες ενώ επίσης μπορεί να επέστρεψε έναν αριθμό τους. Θεωρητικά αναμένουμε πως ο αριθμός που προκύπτει από τη διαφορά των πωληθείσων ποσοτήτων και των επιστραμμένων είναι μεγαλύτερος ή ίσος του μηδενός (αφού κανείς δεν μπορεί να επιστρέψει μεγαλύτερο αριθμό από προϊόντα από όσα αγόρασε). Παρόλα αυτά στην πράξη γνωρίζουμε ότι αυτό είναι μια υπόθεση, της οποίας η αλήθεια θα πρέπει να εξεταστεί. Συνεπώς σκοπός μας είναι για κάθε πελάτη να συμψηφίσουμε τις ποσότητες που αγόρασε και τις ποσότητες που ακύρωσε και, εάν η τελική διαφορά ανά προϊόν βρεθεί με αρνητικό πρόσημο, να διαγραφούν από το dataset όλες οι εγγραφές που αναφέρονται στον συγκεκριμένο συνδυασμό πελάτη και κωδικού προϊόντος.

## 2.3 Ενέργειες μεταχείρισης αρχικού συνόλου δεδομένων εν συνόψει

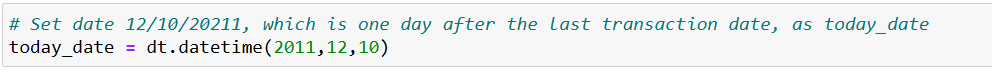
Στην παρακάτω λίστα παρατίθενται συνοπτικά οι αλλαγές που αναφέρθηκαν παραπάνω που είτε έχουν γίνει είτε θα γίνουν για τον καθαρισμό και την προετοιμασία ενός ορθού dataset για το επόμενο βήμα, το οποίο θα είναι η σύσταση του dataset που θα χρησιμοποιηθεί για την ανάλυση RFM:

* Αναφορικά με τα πρωταρχικά δύο υποσύνολα δεδομένων: διαγραφή των εγγραφών των τελευταίων εννέα ημερών από το πρώτο υποσύνολο, και έπειτα συγχώνευση των δύο υποσυνόλων σε ένα ενιαίο
* Μετατροπή των τιμών της στήλης Invoice σε τύπο string
* Μετατροπή των τιμών της στήλης StockCode σε τύπο string, κεφαλαία, και αφαίρεση των περιμετρικών κενών χαρακτήρων
* Διαγραφή ελλειπουσών τιμών από την στήλη Customer ID
* Η διαγραφή των ελλειπουσών τιμών από την στήλη Customer ID ταυτόχρονα συμπαρασύρει και όλες τις εγγραφές οι οποίες έχουν αρνητική τιμή στην στήλη Price, τις οποίες κατά συνέπεια δεν χρειάζεται να διαγράψουμε
* Για κάθε μοναδικό συνδυασμό Customer ID και StockCode άθροιση των ποσοτήτων των επιμέρους τιμολογίων στα οποία αναφέρεται ο κάθε κωδικός προϊόντος StockCode, και μετέπειτα διαγραφή όλων εκείνων των εγγραφών για τις οποίες το προκύπτον άθροισμα είναι αρνητικό.
* Αφαίρεση της ώρας από την στήλη InvoiceDate

Τα παραπάνω βήματα παρουσιάζονται αναλυτικά στο notebook με την ονομασία 01.cleandata[[4]](#footnote-4) το οποίο είναι διαθέσιμο στην ιστοσελίδα GitHub.

# Η μέθοδος RFM

## 3.1 Υπολογισμός των μετρικών Recency, Frequency, Monetary

Για την τμηματοποίηση των πελατών σε ομάδες με όμοια χαρακτηριστικά, διαμορφώνεται το σύνολο δεδομένων RFM. Για τον υπολογισμό της μετρικής Recency θα πρέπει αρχικά να σημειωθεί πότε πραγματοποιήθηκε η τελευταία συναλλαγή του dataset (09/12/2011). Ως ημερομηνία αναφοράς ορίζεται η αμέσως επόμενη ημέρα, δηλαδή η ημερομηνία 10/12/2011. Η μεταβλητή Recency προκύπτει από τον αριθμό των ημερών που μεσολαβούν ανάμεσα στην ημερομηνία της πιο πρόσφατης αγοράς που πραγματοποίησε ο κάθε πελάτης και στην ημερομηνία αναφοράς.

A screenshot of a computer

Description automatically generatedA screen shot of a computer code

Description automatically generatedΗ μετρική Frequency ορίζεται ως ο συνολικός αριθμός μοναδικών τίτλων προϊόντων που αγόρασε ένας πελάτης, δηλαδή ο συνολικός αριθμός εγγραφών κάθε πελάτη στο συγκεκριμένο dataset. Επιπλέον, για τον υπολογισμό της μετρικής Monetary, είναι απαραίτητη η δημιουργία της στήλης TotalValue η οποία προκύπτει από το γινόμενο των στηλών Price και Quantity για κάθε συναλλαγή. Η μετρική Monetary προκύπτει από το άθροισμα των τιμών της στήλης TotalValue για κάθε πελάτη.[[5]](#footnote-5)

Έχοντας ως βάση τις στήλες InvoiceDate, Invoice και TotalValue και με τον συνδυασμό των εντολών της Python groupby() και agg(), υπολογίζουμε τις μετρικές Recency, Frequency και Monetary για κάθε πελάτη.

A screenshot of a computer code

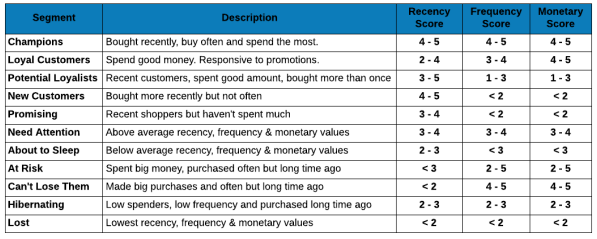
Description automatically generatedΤα παραπάνω βήματα υλοποιήθηκαν αντίστοιχα και στο Power BI όπου με την χρήση των εντολών DATEDIFF, COUNT και SUM της γλώσσας DAX υπολογίστηκαν ως measures οι τρεις μετρικές για κάθε πελάτη.

## 3.2 Τμηματοποίηση πελατών

Ο υπολογισμός των παραπάνω μετρικών για κάθε πελάτη δίνει την δυνατότητα στην επιχείρηση να έχει πρόσβαση σε χρήσιμες πληροφορίες, ο όγκος των οποίων παρόλα αυτά τις καθιστά δύσκολα διαχειρίσιμες. Για να αντιμετωπιστεί αυτό το πρόβλημα, σύμφωνα με την RFM ανάλυση, για κάθε μία από τις μετρικές Recency, Frequency και Monetary, κάθε καταναλωτής λαμβάνει μία βαθμολογία (score) από τον αριθμό 1 έως τον αριθμό 5 (το 5 αντιπροσωπεύει την καλύτερη βαθμολογία). Για να γίνει αυτό, το σύνολο των καταναλωτών χωρίζεται σε 5 ισοπληθείς κατηγορίες. Το 20% των πελατών με μεγαλύτερη συχνότητα συναλλαγών λαμβάνει score 5 για την στήλη Frequency, το 20% των πελατών με τις αμέσως μεγαλύτερες συχνότητες λαμβάνει score 4, και η λογική αυτή επαναλαμβάνεται μέχρι το score 1. Η ίδια διαδικασία επαναλαμβάνεται για τη στήλη Monetary, ενώ για τη στήλη Recency μεγαλύτερα score λαμβάνουν μικρότερες τιμές εγγύτητας αγοράς προς την ημερομηνία αναφοράς. Στην συνέχεια για κάθε πελάτη οι βαθμολογίες που έχει λάβει σε κάθε κατηγορία συνενώνονται και δημιουργούν το τριψήφιο RFM score του πελάτη.

Ο παραπάνω τρόπος εξαγωγής των RFM scores είναι ο προκαθορισμένος τρόπος που αναφέρεται στην βιβλιογραφία. Καθώς όμως όλα τα dataset δεν έχουν τα ίδια χαρακτηριστικά μεταξύ τους δεν επιδέχονται απαραίτητα της ίδιας αντιμετώπισης γενικής χρήσεως. Για παράδειγμα θα μπορούσε να υφίσταται σύνολο δεδομένων για το οποίο είναι αδύνατος ο χωρισμός των πελατών σε 5 ισόποσες κατηγορίες για κάποια από τις τρεις μετρικές, στην οποία περίπτωση θα έπρεπε να χρησιμοποιηθεί κάποιος άλλος τρόπος ομαδοποίησης σε 5 κατηγορίες, όπως η ομαδοποίηση μέσω clustering. Σε αυτό το πρόβλημα θα είχαμε καταλήξει με το παρόν σύνολο δεδομένων εάν για την μετρική Frequency αντί για τον μοναδικό αριθμό τίτλων προϊόντων είχε χρησιμοποιηθεί ο μοναδικός αριθμός τιμολογίων. Επιπρόσθετα, δεν είναι παράλογο για μία εταιρία να επιθυμεί να προσαρμόσει την αντιμετώπιση των πελατών της στα ιδιαίτερα και ιδιάζοντα χαρακτηριστικά των συνόλων δεδομένων της, με αποτέλεσμα να επιθυμεί να τους ομαδοποιήσει με λεπτότερο τρόπο ή διαφορετικά κριτήρια από το χωρισμό τους σε πέντε ισόποσες ομάδες. Για αυτό τον λόγο παρακάτω παρουσιάζουμε δύο διαφορετικές μεθόδους εξαγωγής των RFM scores (οι οποίες μπορεί να απευθύνονται σε αναλυτές διαφορετικού επιπέδου γνώσεων προγραμματισμού), και μία μέθοδο τμηματοποίησης πελατών, η οποία σε αντίθεση με τις άλλες δύο εφαρμόζεται απευθείας στον τρισδιάστατο RFM χώρο.

### Mέσω υπολογισμού των RFM scores τους

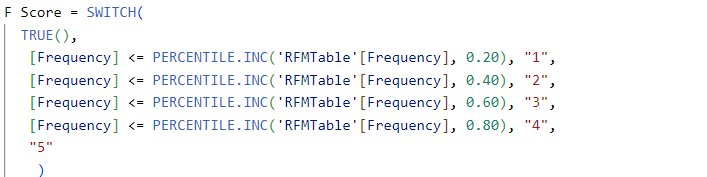
Στην παρούσα ενότητα παρουσιάζονται δύο διαφορετικές μέθοδοι υπολογισμού των RFM scores τα οποία στην συνέχεια θα χρησιμοποιηθούν για τον καθορισμό των τμημάτων (segments) των πελατών της επιχείρησης. Ο συνηθέστερος εν λόγω τρόπος παραγωγής τμημάτων βασίζεται στην εφαρμογή στατικών κανόνων οι οποίοι ομαδοποιούν πελάτες με συναφή RFM scores στα λεγόμενα τμήματα. Ο αριθμός των κανόνων (και συνεπώς η αναλυτικότητα τους) και η συνολική ή μερική κάλυψη του αριθμού των πελατών αποτελεί ανάκλαση της εκάστοτε στρατηγικής προσέγγισης της εταιρίας που εφαρμόζει την RFM ανάλυση, δηλαδή των πόρων της και των επιθυμητών οραματιζόμενων αποτελεσμάτων της διοίκησής της. Στην εικόνα **??** παρατίθεται μία συνήθης τμηματοποίηση σε ένδεκα τμήματα, των οποίων οι κανόνες εξαγωγής με βάση ένα RFM score εμφανίζονται στις τρεις τελευταίες στήλες του εικονιζόμενου πίνακα (Aylanur Cuce).

#### 3.2.1.1 Μέσω πεμπτημορίων

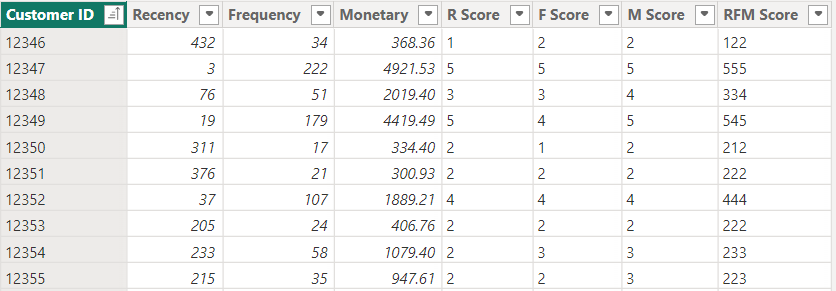
Στις παρακάτω εικόνες παρουσιάζεται ο τρόπος υπολογισμού των R και F scores μέσω Power BI με την χρήση της συνάρτησης PERCENTILE.INC(). Το M score υπολογίζεται κατά τον ίδιο τρόπο με το F score. Στο τέλος τα 3 scores συνενώνονται για την δημιουργία του RFM score κάθε πελάτη.

A screenshot of a computer program

Description automatically generated

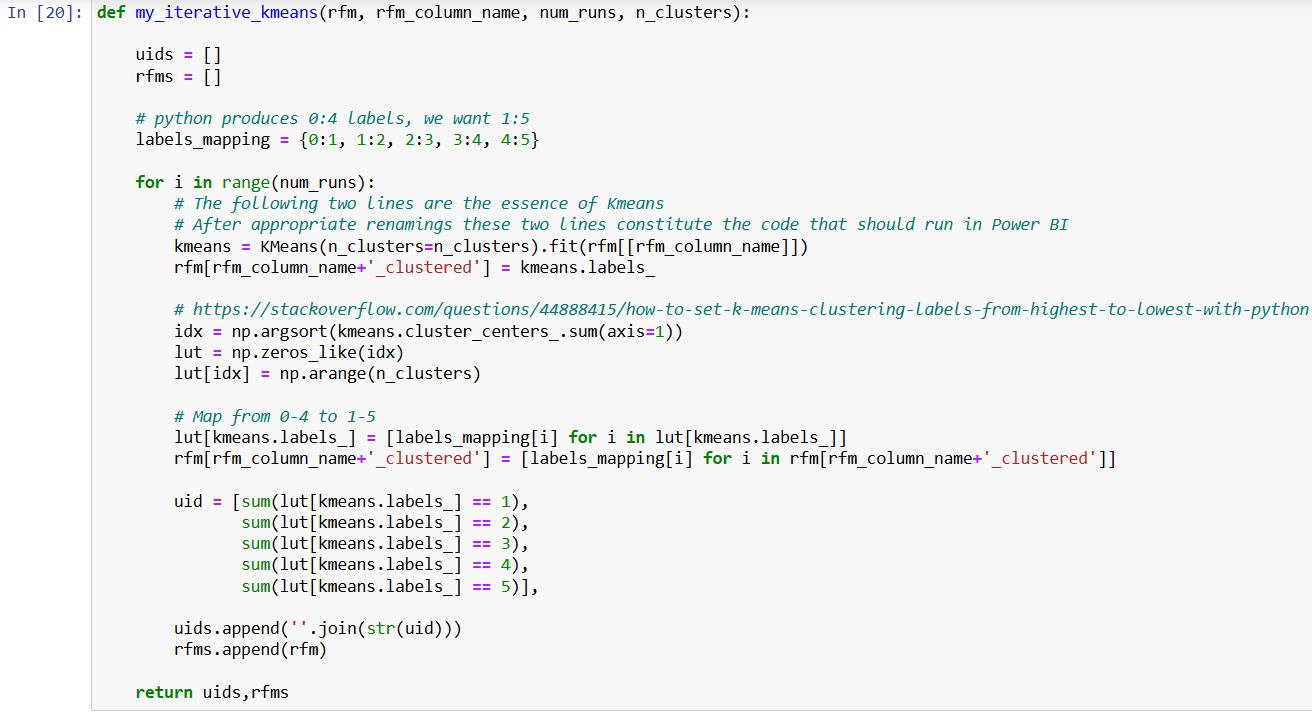




Ο πίνακας **??** απεικονίζει τα RFM scores των 10 πρώτων πελατών ταξινομημένων κατά αύξουσα σειρά με βάση το Customer ID τους.

#### 3.2.1.2 Μέσω συσταδοποίησης ανά στήλη

Στην παρακάτω εικόνα φαίνεται η συνάρτηση στην γλώσσα Python που μπορεί να χρησιμοποιηθεί για να ομαδοποιήσει τα δεδομένα της στήλης rfm\_column\_name του συνόλου δεδομένων rfm σε n\_clusters.



Το αποτέλεσμα του clustering μέσω k-means είναι πιθανοτικό, το οποίο σημαίνει ότι δεν είναι απαραίτητο ότι κάθε εκτέλεσή του παράγει το ίδιο αποτέλεσμα. Για αυτό τον λόγο, διεξάγουμε clustering μέσω k-means num\_runs = 1000 φορές, ώστε να δεχθούμε ως ορθό αποτέλεσμα των RFM scores το πιο συχνό αποτέλεσμα συσταδοποίησης. Σαν επόμενο βήμα θα μπορούσαμε να εισάγουμε αυτό το αποτέλεσμα στο Power BI και να προχωρήσουμε και στα δύο εργαλεία στην περαιτέρω ανάλυση με την λογική RFM, ήτοι στην ομαδοποίηση των πελατών στον τρισδιάστατο RFM χώρο. Παρατηρήστε ότι θα μπορούσαμε να κάνουμε ακριβώς την ίδια ανάλυση εκτελώντας κώδικα Python μέσω Power BI και να εισάγουμε το αποτέλεσμα πίσω στο Jupyter notebook. Αυτός ο τρόπος όμως θα ήταν πιο χρονοβόρος και θα απαιτούσε από αυτόν που χρησιμοποιούσε αποκλειστικά το Power BI ως εργαλείο γνώσεις Python.

Ο πίνακας **??** απεικονίζει τα RFM scores των 10 πρώτων πελατών ταξινομημένων κατά αύξουσα σειρά με βάση το Customer ID τους.

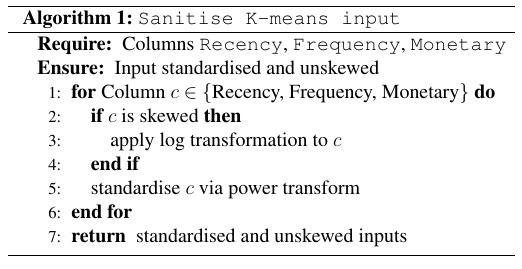
A screenshot of a computer

Description automatically generated

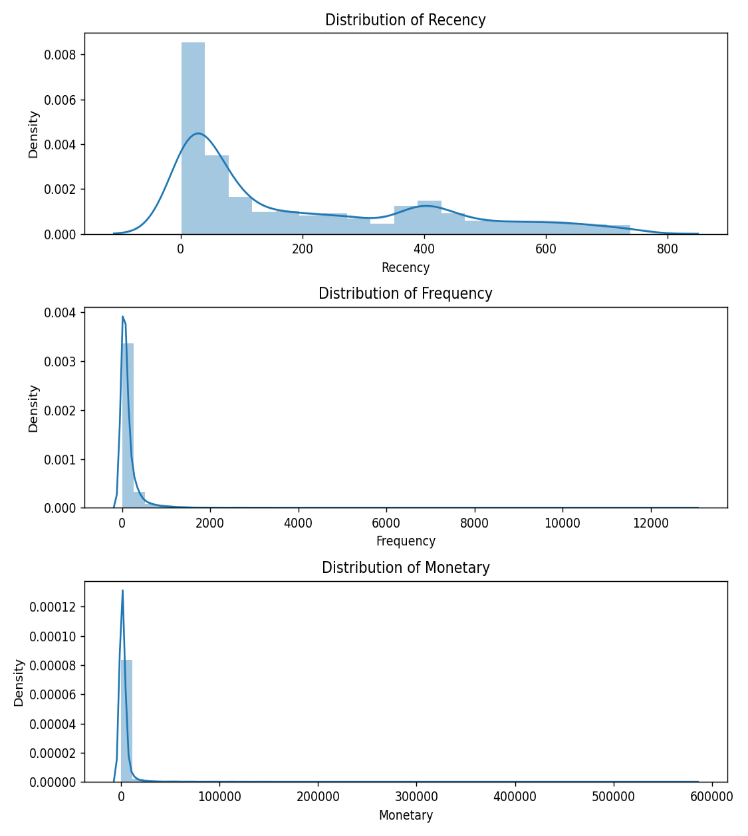
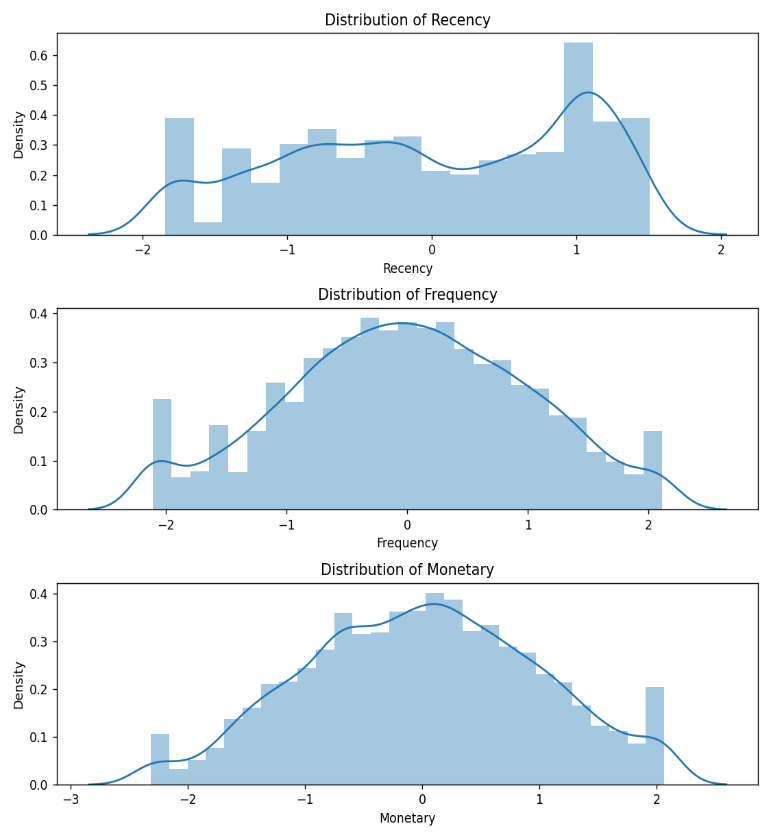
### Μέσω συσταδοποίησης στον τρισδιάστατο RFM χώρο

Στην παρούσα ενότητα παρουσιάζεται η τρίτη μέθοδος τμηματοποίησης των πελατών της επιχείρησης, η οποία δεν απαιτεί τη δημιουργία βαθμολογιών αλλά πραγματοποιείται με την εφαρμογή του αλγορίθμου k-means απευθείας στα σημεία του τρισδιάστατου χώρου που έχει ως διαστάσεις του τις μετρικές Recency, Frequency και Monetary.

Από τη βιβλιογραφία ?? είναι γνωστό πως ο αλγόριθμος k-means παράγει καλύτερα αποτελέσματα όταν τα δεδομένα στα οποία εφαρμόζεται ικανοποιούν τις δύο συνθήκες που παρουσιάζονται στη συνέχεια. Η πρώτη συνθήκη αναφέρεται στον έλεγχο της λοξότητας (skewness) της κατανομής που τα δεδομένα ακολουθούν. Καθώς τα δεδομένα και των 3 μεταβλητών είναι ασύμμετρα κατανεμημένα (κατανομές λοξές δεξιά), αυτά λογαριθμοποιήθηκαν ώστε να μειωθεί η λοξότητά τους (να τείνει όσο γίνεται στο 0). Η δεύτερη συνθήκη που πρέπει να ικανοποιείται αφορά την έκφραση των δεδομένων με όρους κοινής κλίμακας (standardisation). Η κανονικοποίηση είναι η διαδικασία με την οποία επαναπροσδιορίζονται οι τιμές μιας μεταβλητής με βάση την τυπική απόκλισή τους. Η προκύπτουσα τυπική απόκλιση μετά την κανονικοποίηση είναι 1. Από τα ιστογράμματα της εικόνας **??** παρατηρείται ότι τα δεδομένα των στηλών είναι εκφρασμένα σε διαφορετικές μονάδες μέτρησης, και συνεπώς για την προσαρμογή τους σε κοινή κλίμακα έγινε χρήση του PowerTransformer της βιβλιοθήκης sklearn. Στις εικόνες ?? και ?? παρουσιάζονται οι αλγόριθμοι που εφαρμόζονται για την εκτέλεση των παραπάνω βημάτων εκφρασμένοι σε A white box with black text

Description automatically generatedψευδοκώδικα.

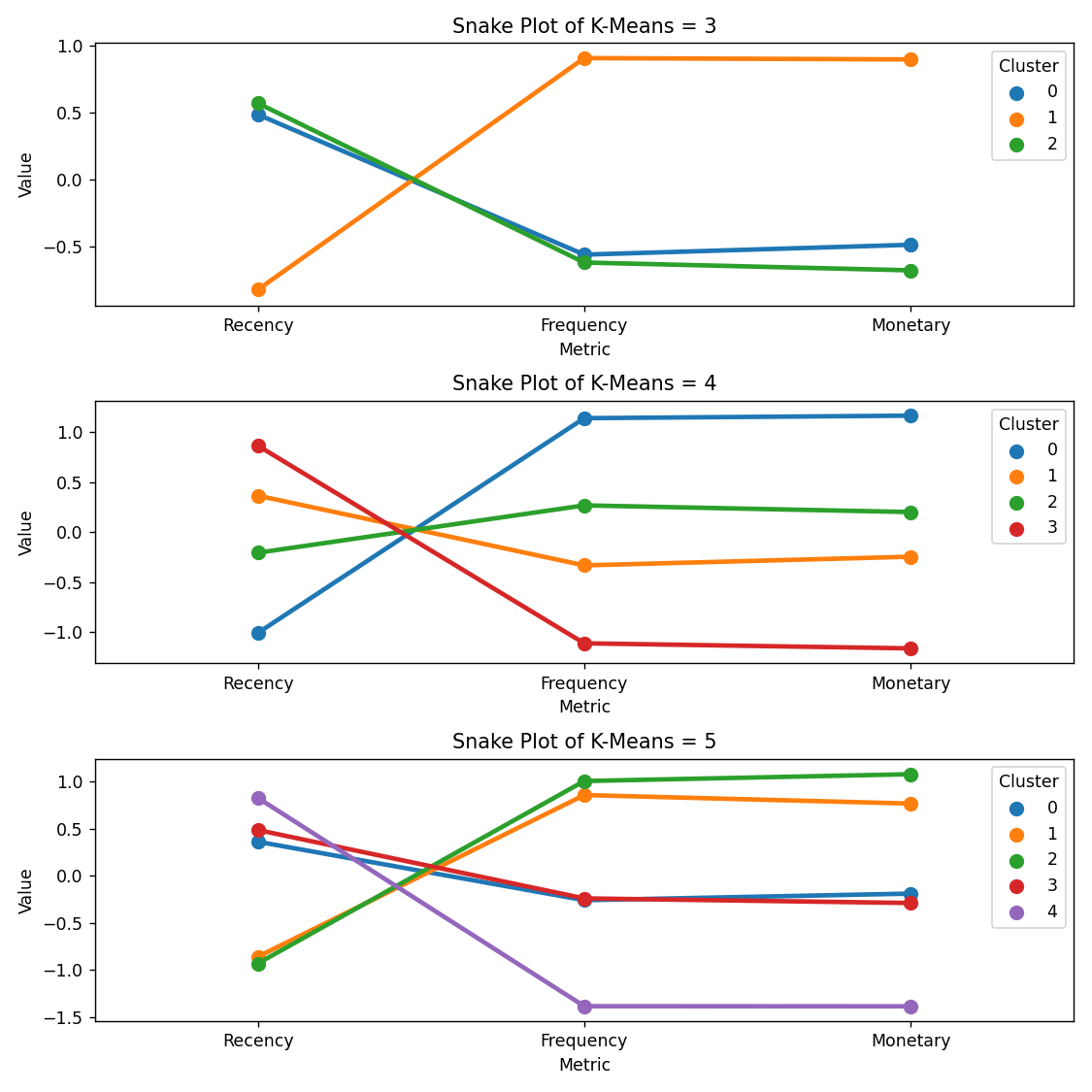
Στην εικόνα **??** παρουσιάζονται οι κατανομές των δεδομένων έπειτα από τους προαναφερθέντες μετασχηματισμούς.



A graph with a line

Description automatically generatedΑφού κανονικοποιηθούν τα δεδομένα, το επόμενο βήμα είναι η επιλογή του βέλτιστου αριθμού συστάδων των δεδομένων στον τρισδιάστατο RFM χώρο. Προς αυτή την κατεύθυνση εφαρμόζουμε τη μέθοδο Elbow,κατά την οποία τρέχουμε τον αλγόριθμο k-means επανειλημμένα για k=1,…,10 και για κάθε τιμή του k υπολογίζουμε την τιμή WCSS (within-cluster sum of squares). Η τιμή WCSS δείχνει ουσιαστικά την «παραμόρφωση» (distortion) που παρατηρείται στην ομοιομορφία των σημείων που σχηματίζουν την κάθε συστάδα: είναι εύλογο να επιθυμούμε την μικρότερη δυνατή «παραμόρφωση». Η μέθοδος Elbow υποδεικνύει ότι το βέλτιστο k εντοπίζεται στο σημείο από το οποίο η «παραμόρφωση» ξεκινά να μειώνεται με γραμμικό τρόπο (εικόνα **??**). Στην συγκεκριμένη περίπτωση παρατηρούμε πως αυτή η συνθήκη ισχύει στη γειτονιά της τιμής k = 4.

Για την εξακρίβωση της ακριβούς ελάχιστης τιμής του αριθμού συστάδων k στρεφόμαστε στη χρήση των snake plots της εικόνας, όπου για k=3, k=4, και k=5, παρουσιάζονται οι μέσες τιμές των μετρικών Recency, Frequency, και Monetary των σημείων που ανήκουν σε κάθε συστάδα, ανά συστάδα. Στα εν λόγω διαγράμματα παρατηρούμε ότι για k=3 και k=5 υπάρχουν συστάδες που με μικρές αποκλίσεις εμφανίζουν τις ίδιες μέσες τιμές μεταξύ τους (πχ για k=3 οι συστάδες 0 και 2, και για k=5 ανά δύο οι συστάδες 1 και 2, και οι 0 και 3). Αυτό το γεγονός υπονοεί πως, κατά μέσο όρο, τα σημεία αυτών των συστάδων ισαπέχουν από τις δύο συστάδες, και συνεπώς ότι η αντιστοίχιση των σημείων στις συστάδες είναι αμφίσημη. Αντιθέτως, για k=4 οι αντίστοιχες μέσες τιμές είναι διακριτές μεταξύ τους ανά συστάδα. Κατά συνέπεια επιλέγουμε να τμηματοποιήσουμε το σύνολο των πελατών σε τέσσερα μέρη.

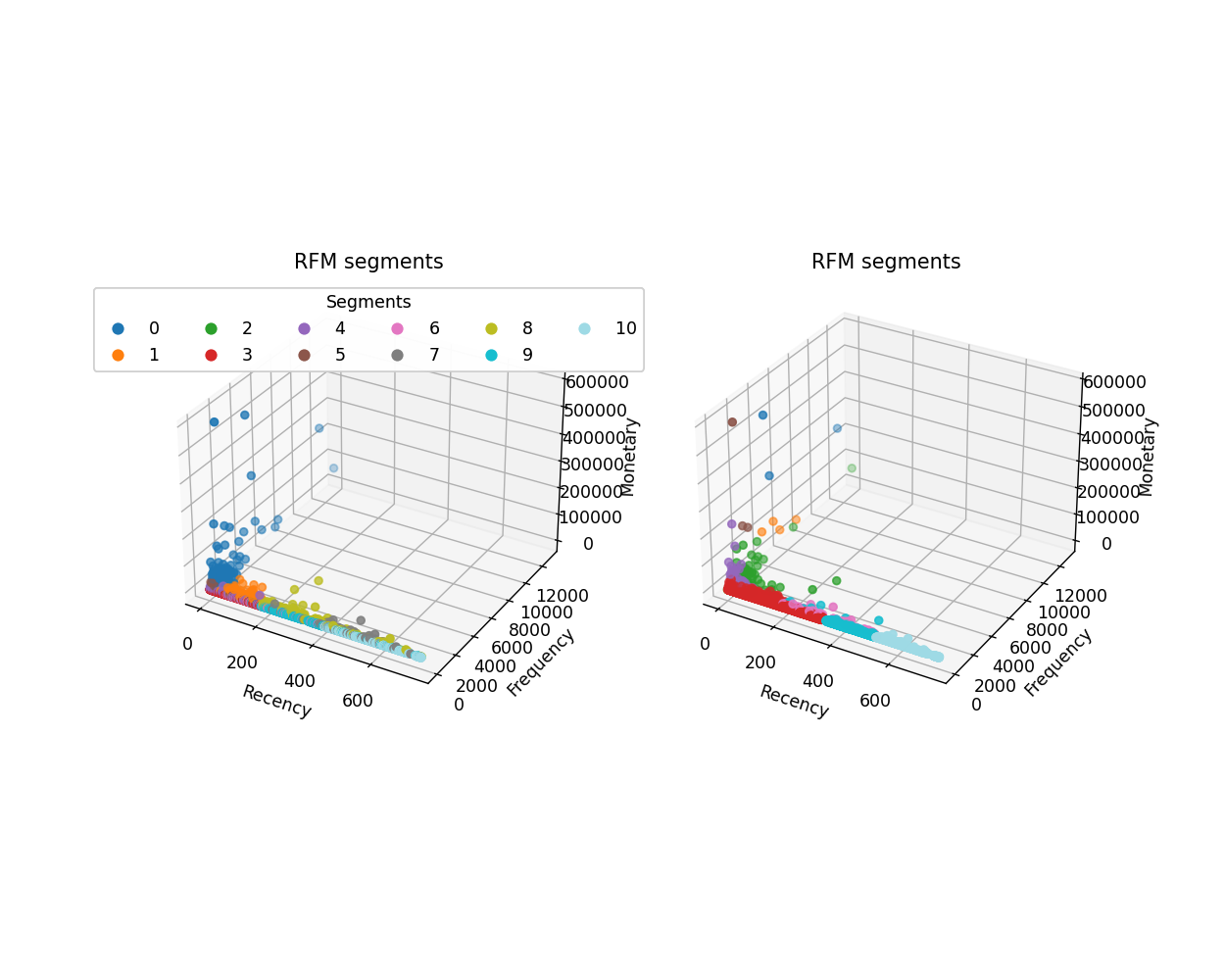


# Αποτελέσματα-Συζήτηση

## 4.1 Αποτελέσματα κατηγοριοποίησης πελατών μέσω percentiles και column k-means

Στην εικόνα ?? παρουσιάζεται στον τρισδιάστατο RFM χώρο η κατηγοριοποίηση των πελατών σε 11 και 9 τμήματα αντίστοιχα με βάση το RFM score τους, έπειτα από την εφαρμογή των δύο διαφορετικών μεθόδων υπολογισμού τους (percentiles και column k-means).

Το κυριότερο που παρατηρούμε στην εικόνα ?? είναι η διαφορά στον χωρικό διαχωρισμό των τμημάτων των δύο μεθόδων. Η μέθοδος column k-means έχει παράξει τμήματα τα οποία είναι περισσότερο ευδιάκριτα σε σχέση με τη μέθοδο percentiles, και με μεγαλύτερη συνοχή μεταξύ τους (τα τμήματα αυτά δεν εμφανίζουν πελάτες άλλων τμημάτων μέσα τους, σε αντίθεση με τη μέθοδο percentiles). Αυτό ίσως οφείλεται στο γεγονός ότι η μέθοδος k-means δεν έχει τον περιορισμό της ομαδοποίησης σε ομάδες με ίσο πλήθος πελατών, όπως η μέθοδος percentiles, δίνοντας έτσι βάση στην χωρική συνάφεια των πελατών και όχι στον απλό ποσοτικό διαχωρισμό τους.



Ε?? Τμηματοποίηση με βάση τη μέθοδο percentiles (αριστερά) και με βάση τη μέθοδο column k-means (δεξιά). Τα ονόματα των τμημάτων δεν αναφέρονται για λόγους οικονομίας χώρου. Για τα ονόματα βλ.Π??

Αποτέλεσμα αυτού του γεγονότος αποτελούν οι αριθμοί των πελατών που περιέχει κάθε τμήμα ανά μέθοδο υπολογισμού, όπως αυτοί απεικονίζονται στην εικόνα ??. Σε αυτό το διάγραμμα βλέπουμε ότι οι αριθμοί πελατών ανά τμήμα στη μέθοδο percentiles εμφανίζουν μικρότερη διακύμανση από αυτούς της μεθόδου k-means, η οποία εν προκειμένω έχει κρίνει πως περίπου το 60% των πελατών του dataset σε σχέση με τους άλλους πελάτες είναι νέοι πελάτες (ανήκουν στο segment id: 03). Αυτό επίσης σημαίνει ότι η μέθοδος column k-means δεν είναι υποχρεωμένη να κατατάξει πελάτες σε κάθε κατηγορία, όπως για παράδειγμα παρατηρούμε ότι δεν εντάσσει κανέναν πελάτη στα segment id: 07 (“cannot lose them but losing”) και segment id: 08 (“at risk”).

A graph with text and numbers

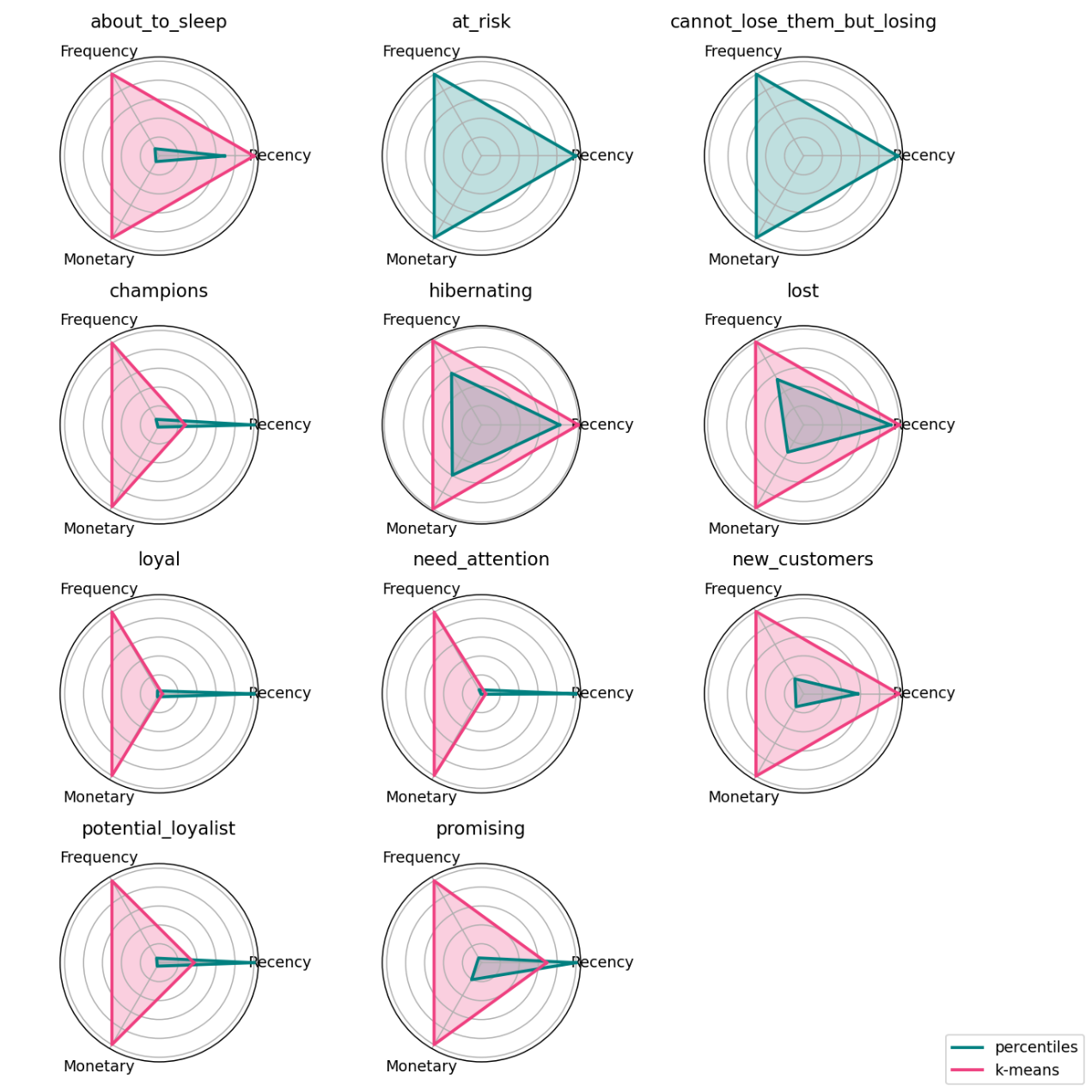
Description automatically generated with medium confidence

Κάτι που αξίζει να σημειωθεί για το συγκεκριμένο σύνολο δεδομένων ως επιπρόσθετο αποτέλεσμα της διαφοράς των δύο μεθόδων ως προς το μέγεθος των πεμπτημορίων/συστάδων είναι η κατανομή των πελατών στα ακραία τμήματα. Παρατηρήστε στην εικόνα ?? πως τα δύο πιο επικερδή τμήματα για την επιχείρηση, δηλαδή οι “champions” και οι “loyal” αποτελούνται από ελάχιστους πελάτες από τη μεν μέθοδο column k-means (3 και 4 αντίστοιχα), ενώ για τη μέθοδο percentiles αυτές οι δύο κατηγορίες αποτελούν σχεδόν το ένα τρίτο των συνολικών πελατών της επιχείρησης. Στον αντίποδα, όσον αφορά στους πελάτες που είναι πλέον αδιάφοροι για την επιχείρηση, οι δύο μέθοδοι βρίσκονται σε συμφωνία ως προς το μέγεθός τους.

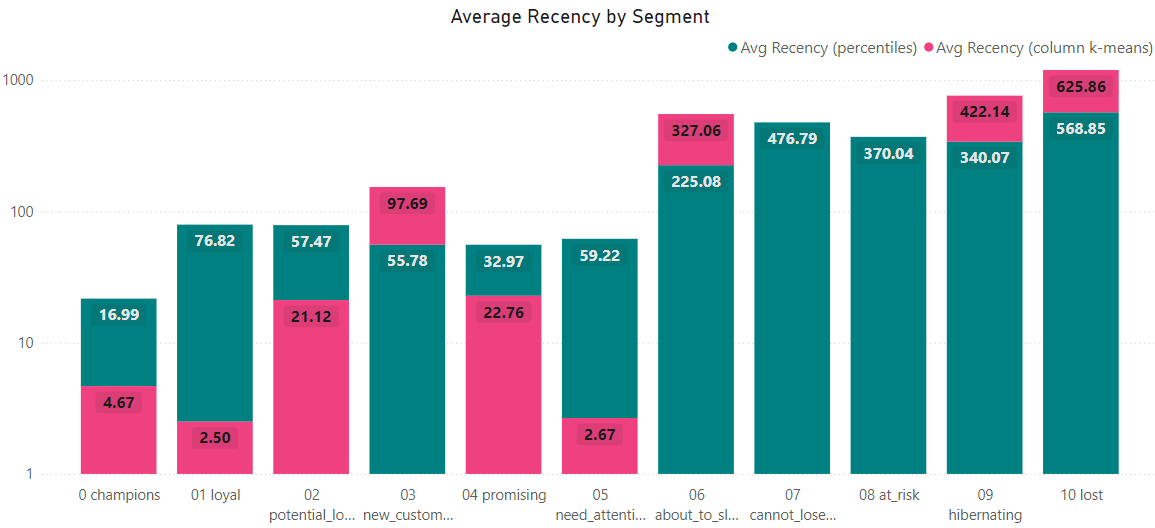
Αν η επιχείρηση αποφασίσει να υιοθετήσει τη μέθοδο **column k-means** για την τμηματοποίηση των πελατών της αυτό πιθανότατα σημαίνει ότι διαθέτει τους πόρους και είναι πρόθυμη να εφαρμόσει μία εξαιρετικά απαιτητική εκστρατεία μάρκετινγκ για την μεγαλύτερη ομάδα πελατών της, τους “new customers” (57,42% του συνόλου των πελατών). Οι νέοι πελάτες χαρακτηρίζονται από πρόσφατες αγορές (υψηλό R), χαμηλή συχνότητα αγορών (χαμηλό F), και μέτριου ύψους χρηματικές δαπάνες (μέτριο M). Στόχος της επιχείρησης είναι να προβιβάσει τους πελάτες αυτού του τμήματος σε “promising” ή και “potential loyalists” και προς επίτευξη του εν λόγω σκοπού απαιτούνται ενέργειες μάρκετινγκ οι οποίες θα την κάνουν να ξεχωρίσει στα μάτια τους σε σχέση με τον ανταγωνισμό. Για να αυξήσει την συχνότητα των αγορών τους μπορεί να τους παρέχει εκπτωτικά κουπόνια για μελλοντικές προσφορές ή μικρά δώρα με την ολοκλήρωση της πρώτης τους αγοράς. Επιπλέον η συμμετοχή τους σε μια έρευνα ικανοποίησης μέσω τηλεφώνου ή ιστοσελίδας όπου θα μπορέσουν να αξιολογήσουν τα προϊόντα, μπορεί να τους κινητοποιήσει σημαντικά προς μελλοντικές αγορές και να τους κάνει να «συνδεθούν» με το brand της επιχείρησης. Όσον αφορά τους “hibernating” πελάτες της που αποτελούν το 2ο μεγαλύτερο segment σύμφωνα με τη μέθοδο column k-means (17,19% των πελατών), συνήθως δεν συνίσταται να επενδύονται πόροι και προσπάθειες προσέγγισης της συγκεκριμένης ομάδας πελατών, καθώς έχει δείξει απροθυμία σε τέτοιου είδους προσπάθειες στο παρελθόν. Αυτοί οι πελάτες είναι οι πελάτες με χαμηλή συχνότητα αγορών, χαμηλές δαπάνες και μη πρόσφατες αγορές. Τέλος, ιδιαίτερο χειρισμό απαιτεί το 3ο σε μέγεθος segment (11,68% των πελατών), οι πελάτες που χαρακτηρίζονται ως “promising”. Οι πελάτες αυτού του τμήματος παρά το γεγονός ότι χαρακτηρίζονται από πρόσφατες αγορές και υψηλές χρηματικές δαπάνες, δεν πραγματοποιούν αγορές σε τακτά χρονικά διαστήματα. Για να αυξηθεί λοιπόν η συχνότητα των αγορών τους (F), η επιχείρηση μπορεί να τους δελεάσει παρέχοντας προτάσεις για παρεμφερή προϊόντα, στέλνοντάς τους μικρά δώρα με κάθε αγορά και ενθαρρύνοντάς τους να κάνουν αξιολόγηση των προϊόντων/υπηρεσιών της επιχείρησης.

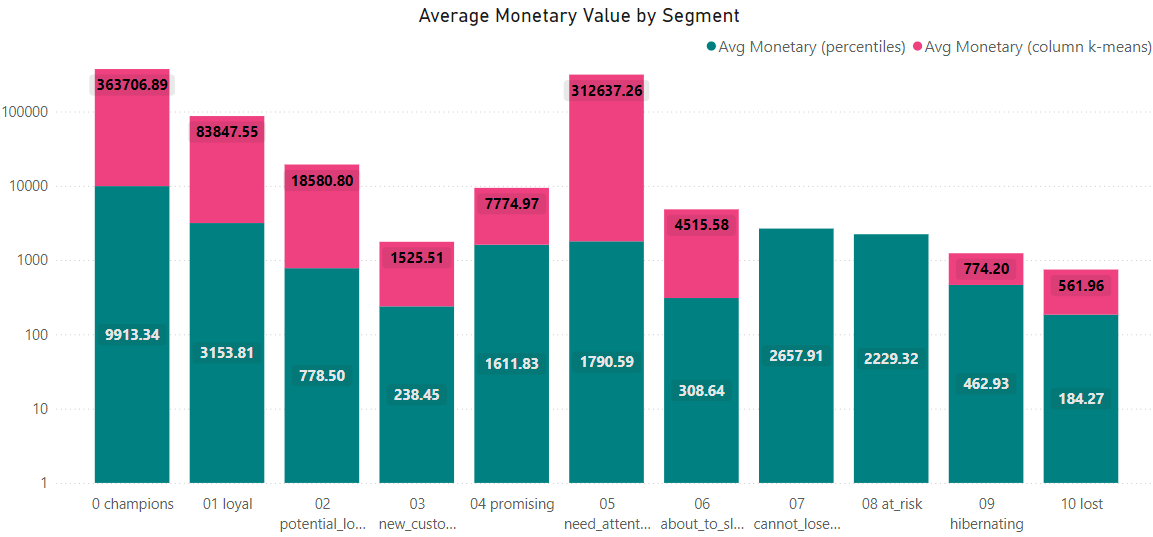
Σε αντίθεση, αν η επιχείρηση αποφασίσει να υιοθετήσει τη μέθοδο **percentiles** για την τμηματοποίηση των πελατών της και τον σχεδιασμό της στρατηγικής μάρκετινγκ, θα πρέπει να στοχεύσει ίσως σε περισσότερα segments και με διαφορετικά είδη ενεργειών, καθώς οι πελάτες είναι πιο ομοιόμορφα κατανεμημένοι στα διάφορα τμήματα. Το μεγαλύτερο σε πλήθος segment που προκύπτει από την εν λόγω μέθοδο είναι οι “hibernating” πελάτες (19,37% των πελατών), ενώ ακολουθούν οι “champions” (19,02% των πελατών) και οι “potential loyalists” (11,01% των πελατών). Όπως προαναφέρθηκε, δεν συνηθίζεται να δαπανώνται πόροι και προσπάθειες για την προσέγγιση των “hibernating” πελατών καθώς παραμένουν απρόθυμοι να επιστρέψουν στην επιχείρηση. Αξίζει όμως να δοθεί έμφαση στην κατηγορία “champions” επιβραβεύοντάς τους καθώς οι πελάτες αυτοί είναι υπεύθυνοι για ένα μεγάλο μερίδιο των εσόδων της αλλά και για την διαφήμιση των προϊόντων της μέσω θετικών σχολίων (positive word of mouth). Η επιχείρηση μπορεί να τους προσφέρει αποκλειστικές προσφορές και προτεραιότητα στη διάθεση νέων προϊόντων. Η επιχείρηση θα πρέπει επίσης να εστιάσει στην προσέγγιση των potential loyalists και συγκεκριμένα στην παρότρυνσή τους να δαπανήσουν περισσότερα χρήματα στα προϊόντα της επιχείρησης, καθώς οι μεταβλητές R και F είναι ήδη υψηλές σε αυτή την κατηγορία των πελατών. Οι προσπάθειες μάρκετινγκ θα πρέπει να επικεντρωθούν σε προτάσεις για συμπληρωματικά ή επιπλέον προϊόντα αλλά και ενέργειες που θα τους κάνουν να νιώσουν πολύτιμοι και να επομένως να αυξήσουν την αφοσίωσή τους στην επιχείρηση.

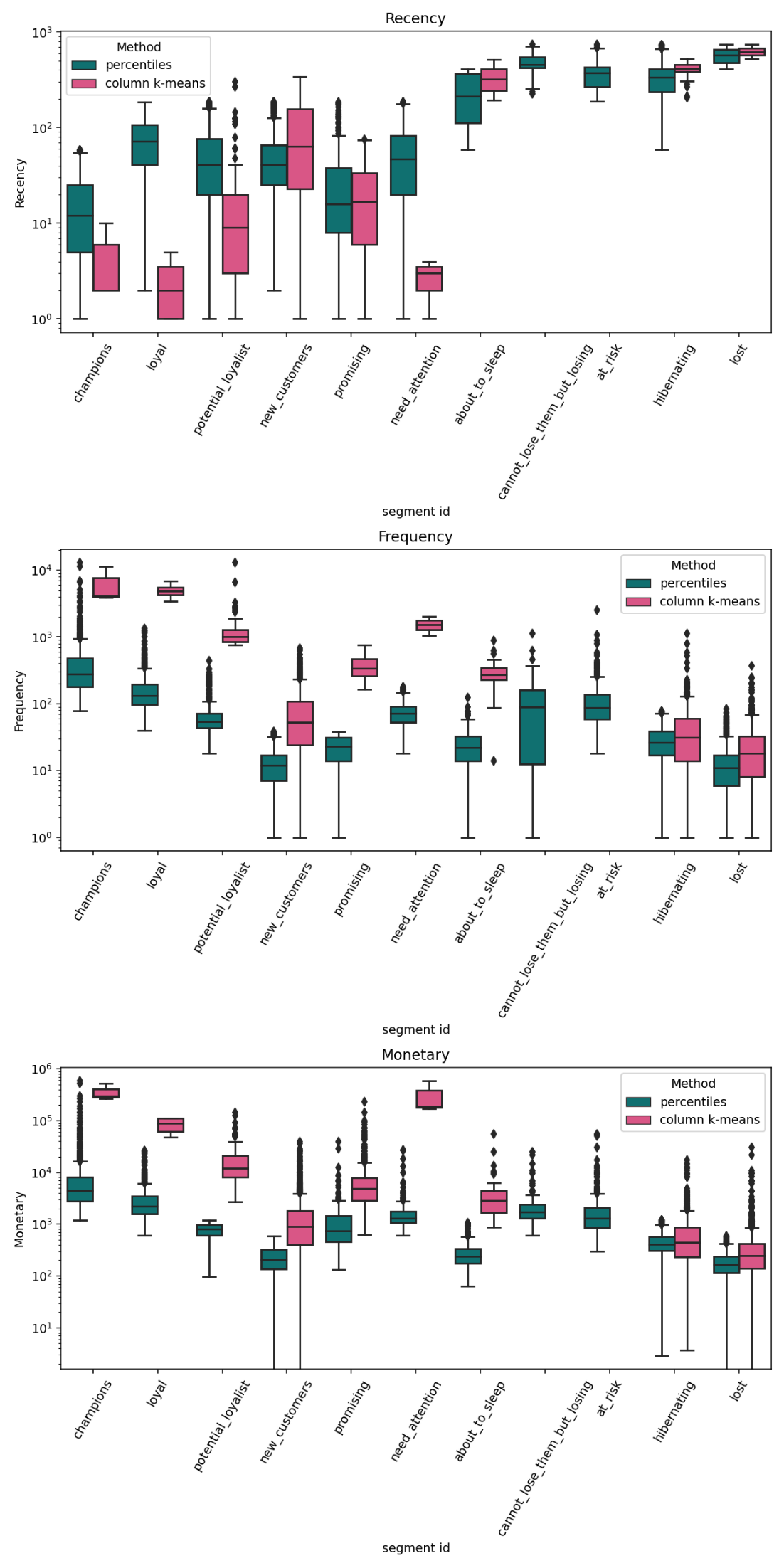
Στο διάγραμμα ?? απεικονίζονται οι ανά τμήμα μέσες τιμές των μετρικών Recency, Frequency και Monetary των δύο μεθόδων ως ποσοστά των εκάστοτε μέγιστων τιμών. Με άλλα λόγια για κάθε τμήμα υπολογίστηκαν οι μέσες τιμές των μετρικών για κάθε μέθοδο και η εκάστοτε μέγιστη τέθηκε ως το 100% επί του οποίου απεικονίζεται η τιμή της μεθόδου που κατέγραψε το ελάχιστο. Εκτός από τις κατηγορίες “hibernating” και “lost” παρατηρούμε και εδώ, με αυτόν τον τρόπο, τις αναντιστοιχίες στην κατανομή των πελατών της κάθε μεθόδου, ποσοτικοποιημένες ανά μετρική ενδιαφέροντος. Στις εικόνες ??, ??, και ?? παρατίθενται σε ραβδογράμματα οι μέσες τιμές των εν λόγω μετρικών ανά τμήμα και ανά μέθοδο, σε λογαριθμική κλίμακα, για μεγαλύτερη λεπτομέρεια.

****

A graph of a bar chart

Description automatically generated with medium confidence



****

Παραδείγματα τμηματοποίησης πελατών

* Οι δύο μέθοδοι συμφωνούν ότι οι 3 καλύτεροι πελάτες της επιχείρησης είναι οι εξής:

A screenshot of a computer

Description automatically generated

~~~~~~~~~~~ ΙΣΩΣ ένα πινακάκι με τις συγκρίσεις των παρακάτω ~~~~~~~~~~~~~~

* Ο πελάτης με Customer ID 17450 πραγματοποίησε την πιο πρόσφατη παραγγελία του 9 ημέρες πριν την ημερομηνία αναφοράς (R) και έχει κάνει συνολικά 418 παραγγελίες (F) των οποίων η συνολική αξία ανέρχεται στις 233803,91 χρηματικές μονάδες. Η μέθοδος percentiles τον κατατάσσει στους “champions” (segment id: 0) με score 555, ενώ η μέθοδος k-means τον κατατάσσει στους “promising” (segment id: 04) με score 524.
* Ο πελάτης με Customer ID 13694 πραγματοποίησε την πιο πρόσφατη παραγγελία του 4 ημέρες πριν την ημερομηνία αναφοράς (R) και έχει κάνει συνολικά 1535 παραγγελίες (F) των οποίων η συνολική αξία ανέρχεται στις 190801,60 χρηματικές μονάδες. Η μέθοδος percentiles τον κατατάσσει στους “champions” (segment id: 0)με score 555, ενώ η μέθοδος k-means τον κατατάσσει στους “need\_attention” (segment id: 05) με score 534.

Η διαφορά στην εν λόγω κατηγοριοποίηση είναι αξιοσημείωτη καθώς υποδηλώνει την διαφορετική αντίληψη/θεώρηση των πελατών και προστάζει δύο πολύ διαφορετικές προσεγγίσεις μάρκετινγκ. Στην περίπτωση των percentiles, ο πελάτης ανήκει στην κατηγορία “champions” και απαιτούνται ενέργειες επιβράβευσής του όπως οφέλη αποκλειστικών προσφορών και early access σε νέα προϊόντα που στο μέλλον θα διαθέτει η επιχείρηση. Εν αντιθέσει στην περίπτωση του k-means, ο πελάτης ανήκει στους “need\_attention”, πελάτες που θεωρείται ότι για την εκ νέου προσέγγισή τους θα πρέπει να χρησιμοποιηθούν ενέργειες εξατομικευμένου μάρκετινγκ όπως υπενθυμίσεις ή προσφορές, ώστε να επιστρέψουν στην επιχείρηση.

* Ο πελάτης με Customer ID 13533 (R=183, F=76, M=270,79) κατατάσσεται στους “about\_to\_sleep” (segment id: 06) με score 331 σύμφωνα με τη μέθοδο percentiles, ενώ η μέθοδος k-means τον κατατάσσει στο τμήμα “new\_customers” (segment: 03).

1. <https://www.kaggle.com/datasets/kabilan45/online-retail-ii-dataset> [↑](#footnote-ref-1)
2. <https://archive.ics.uci.edu/dataset/502/online+retail+ii> [↑](#footnote-ref-2)
3. Η δομή δεδομένων DataFrame είναι η βασική δομή δεδομένων της βιβλιοθήκης pandas. Η βιβλιοθήκη pandas είναι το βασικό μέσο επεξεργασίας δεδομένων σε μορφή πινάκων της Python. [↑](#footnote-ref-3)
4. <https://github.com/ArgyroMp/msc-thesis/blob/master/Notebooks/01.cleandata.ipynb> [↑](#footnote-ref-4)
5. Η ανάλυση που ακολουθεί διεξήχθη και θέτοντας ως Frequency την συνολική ποσότητα προϊόντων που αγόρασε ένας πελάτης. Σε αυτή την περίπτωση οι μετρικές Frequency και Monetary δεν ήταν γραμμικά ανεξάρτητες καθώς και οι δύο περιλάμβαναν τον όρο της συνολικής ποσότητας προϊόντων. Για αυτό το λόγο διαιρέσαμε την μετρική Monetary με την συνολική ποσότητα προϊόντων, όπως είναι ένας δεύτερος τρόπος ανάλυσης RFM που αναφέρεται στην βιβλιογραφία **?**, ώστε στο τέλος να προκύψει η μέση χρηματική αξία που ξόδεψε ο κάθε πελάτης. Στην δική μας περίπτωση ο διαχωρισμός των πελατών με αυτό τον τρόπο δεν απέδωσε αποτελέσματα τόσο σαφή όσο η παραδοσιακή μέθοδος, την οποία ακολουθήσαμε. [↑](#footnote-ref-5)