## **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.

A white background with text

Description automatically generatedΑρχικά 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.

A screenshot of a computer code

Description automatically generatedΠαρακάτω 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:

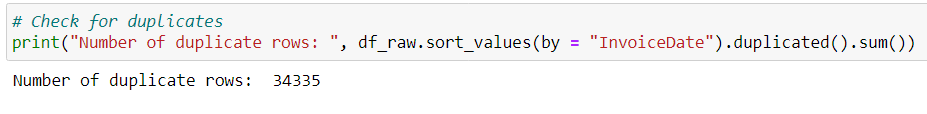
Με την εντολή df\_raw.head(10) βλέπουμε ενδεικτικά τις δέκα πρώτες συναλλαγές του dataset:

A screenshot of a computer screen

Description automatically generated

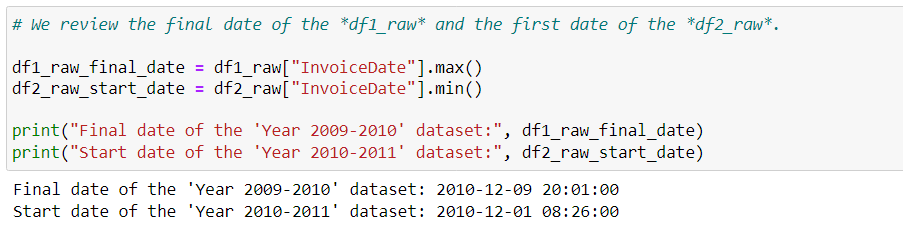
## Καθαρισμός Δεδομένων

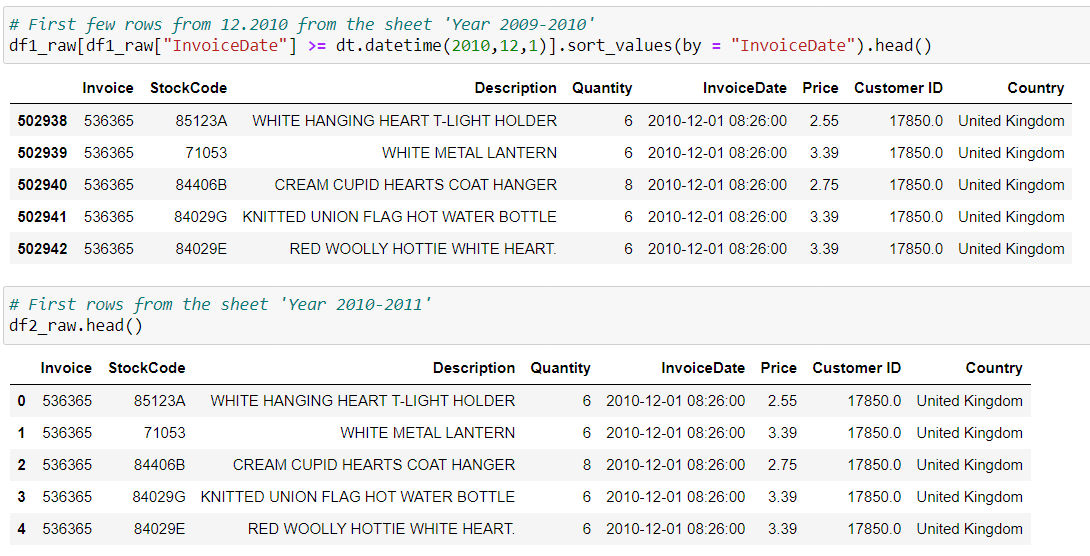
Αφαίρεση διπλών καταχωρίσεων

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.

A graph with blue bars

Description automatically generated with medium confidenceΣτο παρακάτω 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.

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.

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.

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.

A screenshot of a computer code

Description automatically generated

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

A graph with blue bars

Description automatically generated with medium confidence

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 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.

Ελλείπουσες τιμές

Χρησιμοποιώντας την εντολή df.isnull().sum() παρατηρούμε τον αριθμό των ελλειπουσών τιμών σε κάθε μεταβλητή. Οι μόνες μεταβλητές που έχουν missing values είναι η μεταβλητή Description (**4,275**) και η μεταβλητή Customer ID (**235,287**). Για την υλοποίηση της RFM ανάλυσης οι ελλείπουσες τιμές της μεταβλητής Customer ID θα αφαιρεθούν από το dataset.

Προτού ξεκινήσουμε την διαμόρφωση του συνόλου δεδομένων για τις ανάγκες της παρούσας ανάλυσης, θα πρέπει να διερευνηθούν οι μεταβλητές του και να αντληθούν κάποιες χρήσιμες αρχικές πληροφορίες.

Exploratory Data Analysis

Ο μοναδικός αριθμός των τιμολογίων είναι 53,628.

Στήλη Invoice

Από την περιγραφή του συνόλου δεδομένων πληροφορηθήκαμε ότι ορισμένα από τα τιμολόγια αφορούν ακυρώσεις τιμολογίων, συγκεκριμένα όσα ξεκινούν με το γράμμα “C”. Θα προσπαθήσουμε αρχικά να δούμε τον συνολικό αριθμό των ακυρώσεων και έπειτα πώς συνδέονται με τις μεταβλητές Quantity και Price. Για τον σκοπό αυτό, δημιουργήθηκε ένα DataFrame με τις συναλλαγές που αφορούν ακυρώσεις τιμολογίων. Ο αριθμός τους είναι **19,165** που αντιστοιχεί σε ποσοστό 1,8% του συνολικού αριθμού των συναλλαγών του dataset.

A computer code with text

Description automatically generated with medium confidence

Από τις ακυρώσεις τιμολογίων, μόνο μία (1) συναλλαγή παρουσιάζει θετική τιμή στην στήλη Quantity, ενώ οι υπόλοιπες 19,164 έχουν αρνητική τιμή στην ποσότητα. Η εν λόγω συναλλαγή έχει ελλείπουσα τιμή στην στήλη Customer ID.

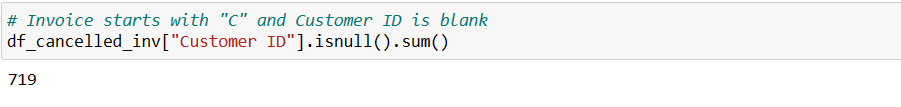
A screenshot of a computer

Description automatically generated

Εκτός από τα τιμολόγια που ξεκινούν από “C” υπάρχουν και κάποια που ξεκινούν από “A”. Τα τιμολόγια αυτά είναι μόνο 6, έχουν missing value στην στήλη Customer ID και τα 5 από αυτά σχετίζονται με αρνητικές τιμές στην στήλη Price (πολύ μεγάλη απόλυτη τιμή) και μοναδιαία τιμή στην στήλη Quantity. Υποθέτουμε ότι πρόκειται για ενέργειες λογιστικής φύσεως και δεν σχετίζονται με τις πωλήσεις του ηλεκτρονικού καταστήματος. Επιπλέον, παρατηρούμε ότι υπάρχει μία συσχέτιση μεταξύ των τιμολογίων “A563185”, “A563186” και A screenshot of a computer

Description automatically generated“A563187”.

Από τα 19,165 τιμολόγια που ξεκινούν με το γράμμα “C”, τα **719** (3%) έχουν missing value στην στήλη Customer ID.



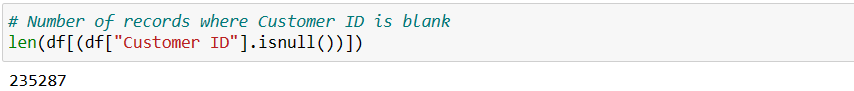
Στήλη Quantity

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 κάποιων προϊόντων στο κατάστημα (Quantity < 0).

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 (19,164). Αυτό υποδηλώνει ότι η αρνητική τιμή στην ποσότητα δεν σημαίνει απαραίτητα μόνο την επιστροφή κάποιου προϊόντος, αλλά μπορεί να είναι συνδεδεμένη και με άλλα είδη συναλλαγών του ηλεκτρονικού καταστήματος, όπως ακυρώσεις πωλήσεων και ελλαττωματικά προϊόντα.

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.

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, θα μπορούσαμε να υποθέσουμε ότι οι συγκεκριμένοι πελάτες δεν πραγματοποίησαν ποτέ εγγραφή στο online κατάστημα και επομένως δεν τους αποδόθηκε αναγνωριστικό πελάτη.

Ο μοναδικός αριθμός των Customer ID που έχουμε στην διάθεσή μας είναι **5,942**.

A close-up of a computer code

Description automatically generated

Στήλη Country

A screenshot of a computer

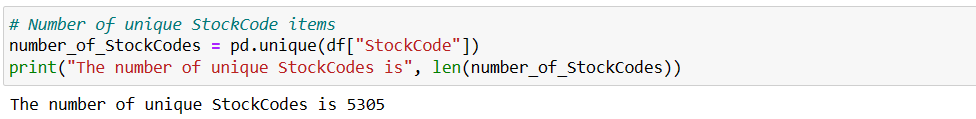
Description automatically generatedΟι συναλλαγές του dataset προέρχονται από 42 χώρες, και ενώ δεν παρατηρούνται missing values στην στήλη Country, υπάρχουν 756 εγγραφές στις οποίες η χώρα είναι “Unspecified”.

A graph with numbers and a bar

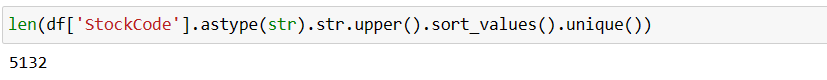
Description automatically generatedΤο μεγαλύτερο μερίδιο των συναλλαγών πραγματοποιήθηκε στο Ηνωμένο Βασίλειο (91.87%), ενώ ακολουθούν η Ιρλανδία (1.69%) και η Γερμανία (1.66%). Στο παρακάτω γράφημα παρουσιάζονται οι 10 χώρες με τον μεγαλύτερο αριθμό συναλλαγών.

Στήλη StockCode

Η στήλη StockCode αναφέρεται στους κωδικούς προϊόντων που αγοράστηκαν. Θα ήταν χρήσιμο να ερευνηθεί ποιοι κωδικοί έχουν τις μεγαλύτερες πωλήσεις κατά το χρονικό διάστημα που εξετάζεται. Προς αυτόν τον σκοπό κλήθηκε η εντολή df.unique() της Python, σύμφωνα με την οποία οι μοναδικές τιμές της στήλης StockCode είναι **5,305**.



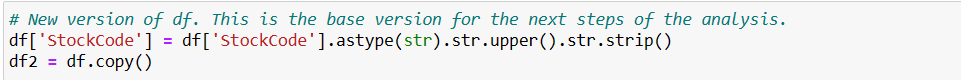
Καθώς όμως τρέξαμε παράλληλα την ίδια εντολή στο Power BI χρησιμοποιώντας την έκφραση DISTINCTCOUNTNOBLANKS της γλώσσας DAX διαπιστώσαμε ότι ο μοναδικός αριθμός προϊόντων διαφέρει, είναι 5,131. Είναι κρίσιμο να μάθουμε γιατί είναι διαφορετικά τα αποτελέσματα που προκύπτουν από τα δύο εργαλεία, και να φέρουμε τις τιμές σε συμφωνία, γιατί αλλιώς η ανάλυση με Python και η ανάλυση με Power BI θα αποκλίνουν και θα αποφέρουν πιθανώς διαφορετικά συμπεράσματα και συνεπώς αμφισημία/αβεβαιότητα.

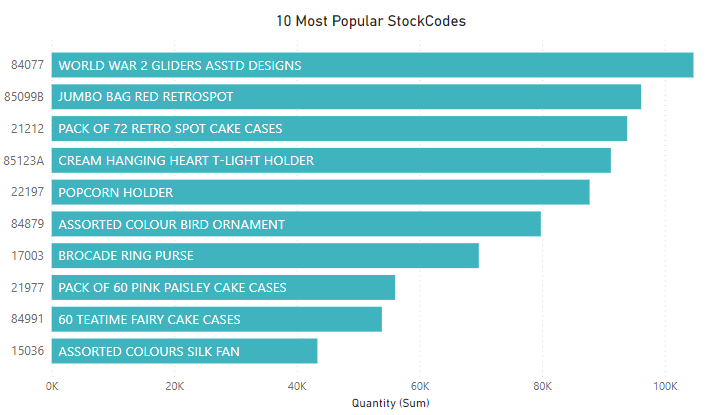
Είναι λογικό να υποθέσουμε ότι η Python διάβασε κάποιες τιμές της στήλης ως μοναδικές, κάτι που δεν έκανε το Power BI, διαφορετικά δεν θα υπήρχε αυτή η διαφορά. Με σκοπό να ερευνήσουμε αν η υποψία αυτή είναι σωστή, πρέπει αρχικά να φέρουμε τις τιμές της στήλης σε ένα κοινό επίπεδο ώστε να είναι συγκρίσιμες μεταξύ τους. Συγκεκριμένα, μετατρέπουμε το data type των τιμών σε 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 μετρήσαμε εκ νέου τις μοναδικές τιμές της στήλης και το αποτέλεσμα ήταν 5,132.

A screenshot of a computer code

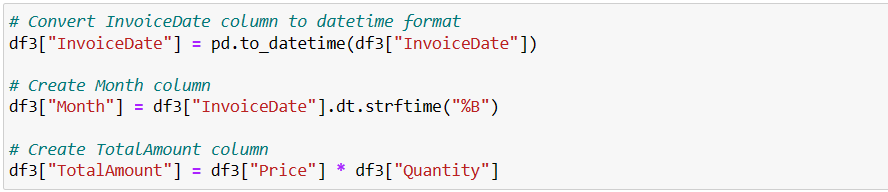
Description automatically generatedΓια να βρούμε την μία τιμή που πλέον εμφανίζεται ως διαφορά, κάναμε export σε excel από την Python και από το PowerBI τις στήλες με τα unique values που προέκυψαν από το κάθε εργαλείο. Κάναμε sorting τις τιμές τους και εντοπίσαμε ότι η εγγραφή 47503J εμφανιζόταν με δύο τρόπους στα αποτελέσματα της Python "47503J " και "47503J". Χρησιμοποιήσαμε την εντολή str.strip() για να αφαιρέσουμε το κενό που δημιουργήθηκε στην ονομασία και τέλος, διαπιστώσαμε ότι οι μοναδικές τιμές της στήλης StockCode είναι 5,131.

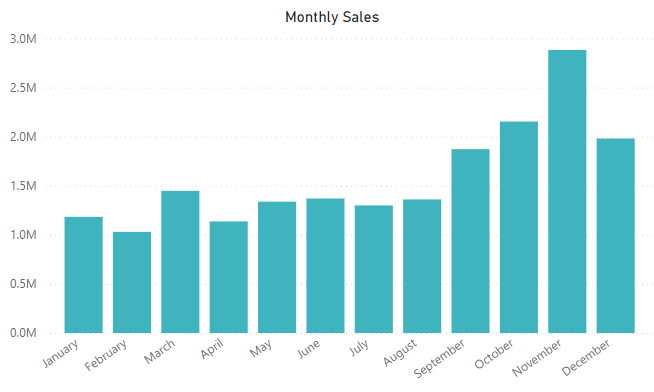
Στο Power BI δεν χρειάστηκε να κάνουμε όλα τα παραπάνω βήματα για να καταλήξουμε στον ίδιο αριθμό, καθώς η εντολή DISTINCTCOUNTNOBLANKS έκανε έναν αριθμό από υποθέσεις, οικεία βουλήσει, αγνοώντας για παράδειγμα τα πεζά και τα κεφαλαία γράμματα ή τους προπορευόμενους και ?? κενούς χαρακτήρες στις ονομασίες 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 υποδήλωνε δύο διακριτά αντικείμενα. 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.

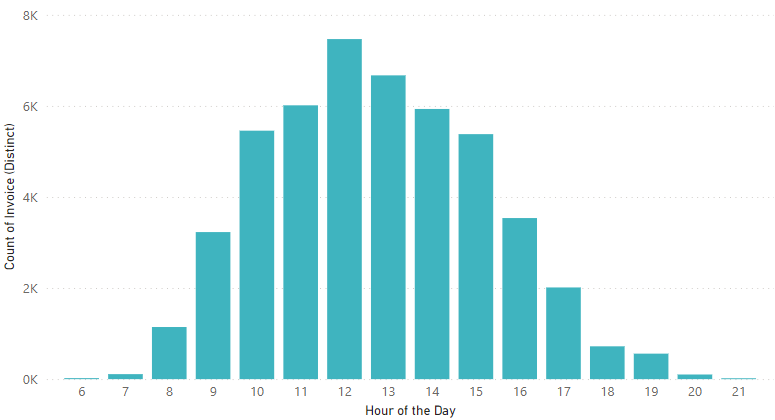
Για να υπολογιστούν λοιπόν τα 10 πιο δημοφιλή προϊόντα του dataset, πρέπει να αποκλειστούν τα προϊόντα που έχουν μηδενική τιμή στην στήλη Price, καθώς τα συγκεκριμένα είναι προϊόντα που δεν πωλήθηκαν αλλά προσφέρθηκαν ως δώρα ή προσφορές από το ηλεκτρονικό κατάστημα, και συνεπώς δεν υποδηλώνουν τις προτιμήσεις των πελατών.

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. Για να ερευνηθεί ποιους μήνες το ηλεκτρονικό κατάστημα έκανε τις περισσότερες πωλήσεις, δημιουργήθηκαν δύο νέες στήλες 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.

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

Επιπλέον, δημιουργήθηκε η στήλη **Hour** για να 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.



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)