Κωνσταντίνος Ανδρέου AM:4316

Η άσκηση υλοποιήθηκε σε γλώσσα προγραμματισμού Java και αποτελείται από 2 διαφορετικές κλάσεις. Μία κλάση Range που αναπαριστά ένα διάστημα τιμών που καταλαμβάνει ένα bin καθώς και τον αριθμό των δεδομένων που πέφτουν σε αυτό και μία κλάση Histogram όπου περιέχει τις κύριες μεθόδους για την δημιουργία και τις λειτουργίες των δυο ιστογραμμάτων.

Η κλάση Range.java αποτελείται από έναν constructor όπου με βάση τις παραμέτρους lowerLimit ,upperLimit και numtuples ορίζει το εκάστοτε bin. Επίσης, περιέχει getters/setters για τα πεδία και μία μέθοδο getInterval() όπου επιστρέφει την διαφορά του άνω με του κάτω ορίου.

Στην κλάση Histogram.java αρχικά ορίζονται ως πεδία το csv αρχείο από το οποίο θα πάρουμε τα δεδομένα, το γνώρισμα(Income) πάνω στο οποίο θα κρατήσουμε τις τιμές του για την παραγωγή των ιστογραμμάτων και ο αριθμός των bins(100). Η βασική μέθοδος του προγράμματος είναι η produceHistograms στην οποία αρχικά ανοίγεται το αρχείο και διαβάζεται η πρώτη γραμμή του, τοποθετώντας τα πεδία που χωρίζονται με κόμμα σε ένα πίνακα. Εν συνέχεια καλώντας την μέθοδο **findFieldPosition**() παίρνουμε την θέση στην οποία βρίσκεται το γνώρισμα(Income) και υστέρα διαβάζουμε το αρχείο μέχρι το τέλος του, τοποθετώντας για κάθε γραμμή για την οποία η θέση του γνωρίσματος που βρήκαμε παραπάνω είναι double σε ένα arraylist. Αφού έχουμε μαζέψει τα δεδομένα σορταρουμε την λίστα και καλούμαι τις μεθόδους **produceEquiWidth** και **produceEquiDepth.**

Η πρώτη μέθοδος παράγει τα εύρη τιμών για το equi width ιστόγραμμα υπολογίζοντας αρχικά την μέγιστη και ελάχιστη τιμή από τα δεδομένα και το διάστημα που θα καταλαμβάνει το κάθε bin με τον τύπο `interval = (maxValue - minValue) / bins;` .Στη συνέχεια, μέσα σε μια for δημιουργούνται τα 100 bins ανά interval διάστημα και μετράμε τα numtuples ανά διάστημα.

Η δεύτερη μέθοδος παράγει τα εύρη τιμών για το equi depth ιστόγραμμα υπολογίζοντας αρχικά των αριθμό των numtuples που θα περιέχει το κάθε range με τον τύπο `numtuples = fieldValues.size() / bins;` και υστέρα δημιουργούμε τα ranges ανά numtuples. Στην περίπτωση όπου περισσέψουν δεδομένα τότε δημιουργώ ένα επιπλέον bin όπου μπαίνουν τα περισσευούμενα.

Για το μέρος 2 της άσκησης δημιούργησα μια μέθοδο **computeActualResults** η οποία παίρνει ως παράμετρο τα δεδομένα του γνωρίσματος, ένα κάτω οριο (α) και ένα άνω οριο (β) και υπολογίζει τον πραγματικό αριθμό τον δεδομένων που πέφτουν σε αυτό το διάστημα. Επίσης, μια μέθοδο **computeEstimationResult** η οποία παίρνει ως παράμετρο μια ArrayList<Range> με τα ranges που

δημιουργούνται από ένα ιστόγραμμα, καθώς και το α και β και εκτιμάει τον αριθμό των δεδομένων μέσα σε αυτό το εύρος.

Συμπεράσματα:

Για τα δεδομένα του αρχείου μας το equi-width ιστόγραμμα έδινε πιο προσεγγιστικά αποτελέσματα στο μεγαλύτερο πλήθος των ερωτημάτων σε σχέση με το equi-depth ιστόγραμμα. Αυτό συμβαίνει επειδή ένα equi-width ιστόγραμμα διασφαλίζει ότι όλοι οι κάδοι έχουν το ίδιο πλάτος, επομένως παρέχει μια πιο ομοιόμορφη αναπαράσταση των δεδομένων σε ολόκληρο το εύρος τους. Αντίθετα, ένα ιστόγραμμα equi-depth μπορεί να έχει ως αποτέλεσμα ορισμένοι κάδοι να έχουν πολύ μεγαλύτερο ή πολύ μικρότερο εύρος από άλλους, γεγονός που μπορεί να οδηγήσει σε ανακρίβειες στις προβλέψεις ερωτημάτων εύρους. Έτσι, για τα δεδομένα του αρχείου όπου τα δεδομένα ακολουθούσαν μια πιο ομοιόμορφη κατανομή το equi-width είναι καταλληλότερο. Για την επαλήθευση αυτού δημιούργησα και μια μέθοδο **benchmark** η οποία επαναληπτικά για 100 επαναλήψεις δημιουργεί τυχαία ranges(α και β) και υπολογίζει την απολυτή διαφορά μεταξύ της εκτίμησης με χρήση ιστογράμματος και του πραγματικού αποτελέσματος και σε κάθε επανάληψη νικητήριο ιστόγραμμα είναι αυτό με την μικρότερη διαφορά, αυξάνοντας τον μετρητή νικών του. Ένα παράδειγμα κλήσης της μεθόδου φαίνεται παρακάτω :

```
Benchmarking the two different Histograms(100 Iterations):
EquiWidth Histogram was the winner!Wins: 86
```