

Project 1 – Bash Scripting

COMP211

ΣΤΑΥΡΟΥ ΟΔΥΣΣΕΑΣ

2018030199

Technical University Of Crete - Oct. 2019

Το 1ο project είχε ως απαιτήσεις να δημιουργήσουμε 2 scripts σε γλωσσά bash.

REGR:

Το σκριπτάκι με όνομα regr υπολογίζει τους συντελεστές γραμμικής παλινδρόμησης a, b, c δηλαδή τη βέλτιστη ευθεία ($cY = aX + b$) που προσεγγίζει όσο καλύτερα τα σημεία που δίνονται ως είσοδος. Τα αρχεία που περιέχουν τα σημεία περνιούνται ως παράμετροι κατά την κλήση του προγράμματος και τυπώνει τα εξής

FILE: xxxxxx, a=xx b=xx c=xx err=xx

Στις ακραίες περιπτώσεις όπου η ευθεία είναι της μορφής $x = C$ (δηλαδή όλα τα σημεία έχουν το ίδιο X), το a παρουσιάζεται ως ND (not-defined) διότι η κλίση δεν προσδιορίζεται και το b είναι 0 διότι δεν τέμνει τον y'y πουθενά.

Το regr λειτουργά ως εξής

- Για κάθε αρχείο ως παράμετρος
- Αλλάζει το IFS σε “:” και διαβάζει κάθε γραμμή εκχωρώντας τις 2 τιμές X, Y σε 2 λίστες ανάλογα.
- Διασχίζει την κάθε λίστα ξεχωριστά και υπολογίζει τα εξής αθροίσματα.

$$\text{sum_x} = \sum_i X[i], \text{sum_x2} = \sum_i X[i]^2, \text{sum_y} = \sum_i Y[i]$$

$$\text{sum_xy} = \sum_i X[i]*Y[i]$$

- Αφού έχει δημιουργήσει τα παραπάνω αθροίσματα τότε υπολογίζει τις σταθερές

$$a =, b = \frac{\text{sum_y} - a*\text{sum_x}}{\text{length}}$$

Το length είναι ο αριθμός των σημείων που δίνονται (αριθμός γραμμών στο αρχείο), παράγεται από την εντολή

length=\$(grep ^ -c \$arg)

- Όπως προαναφέρθηκε πιο πάνω όταν τα σημεία είναι στο ίδιο X τότε το a δεν προσδιορίζεται διότι ο παρονομαστής θα ήταν 0

- Το σφάλμα err υπολογίζεται με τον εξής τύπο: $\sum_i (Y[i] - (a*X[i] + b))^2$

Όλες οι μαθηματικές πράξεις γίνονται με την εντολή:

bc <<< scale=2;" " όπου εκχωρούμε μέσα την πράξη ως string, το αποτέλεσμα είναι με 2 δεκαδικά ψηφία αν είναι δεκαδικός

- Γίνεται ο έλεγχος αν ο παρονομαστής του a είναι 0, εάν είναι τότε τυπώνετε απευθείας το αποτέλεσμα και προχωράμε στην επομένη παράμετρο.
- Χρησιμοποίησα την εντολή `awk` και `printf` για να τυπώσω το 0 μπροστά από ένα δεκαδικό αριθμό < 0 και επαναφέρουμε το IFS στην κανονική του τιμή.
- Εκτυπώνω στην οθόνη με τη χρήση του `echo` τις σταθερές τις οποίες υπολόγισα και επαναλαμβάνω τη διαδικασία για τα υπόλοιπα αρχεία

Σημείωση: Το πρόγραμμα μου έχει μέσα 3 επαναληπτικές ροές οι οποίες υπολογίζουν τα 4 αθροίσματα. Εφόσον οι ληστές που περιέχουν X , Y είναι ίδιου μεγέθους θα μπορούσα να κάνω μια μόνο ροή και να αθροίσω τα συγκεκριμένα στοιχεία κάθε αθροίσματος. Ωστόσο, επέλεξα να το κάνω διακριτά διότι μαθηματικά κάθε άθροισμα είναι μια επαναληπτική δομή.

RESULTS:

Το δεύτερο σκριπτάκι με τίτλο `results` παίρνει ως είσοδο 1 αρχείο που περιέχει μέσα με την εξής μορφή, τον αγώνα μεταξύ 2 ομάδων και το αποτέλεσμα του αγώνα: `team1-team2:score1-score2`. Πρέπει εμείς να υπολογίσουμε την τελική κατάταξη των ομάδων και ποσά γκολ σκόραραν υπέρ/κατά τους.

Το `results` λειτουργά ως εξής

- Αλλάζει το IFS σε ":"
- Δημιουργεί κάποιες κενές λίστες τις οποίες θα χρειαστούμε αργότερα, και βεβαιώνεται ότι υπάρχει το αρχείο `"out_teams"` προτού το διαγράψει, για να μην προσπαθήσει να διαγράψει ένα αρχείο το οποίο δεν υπάρχει.
- Στη συνέχεια διαβάζει κάθε γραμμή του αρχείου και χωρίζει τους αγώνες από τα σκορ και τα προσθέτει σε 2 λίστες οι οποίες είναι παράλληλες (το 1ο στοιχείο της λίστας με τους αγώνες έχει σκορ το 1ο στοιχείο της λίστας με τα σκορ)
- Ακολούθως αλλάζει ξανά το IFS σε "-" και διασχίζει τη λίστα με τους αγώνες, χωρίζοντας έτσι της ομάδες, αφού εκχωρήσει κάθε ομάδα στον πίνακα `"teams"` τότε ταξινομεί τη λίστα διαγράφοντας έτσι διπλές καταχωρίσεις.
- Μετά, δημιουργεί 3 λίστες όσο το μέγεθος του νέου πίνακα με τις μοναδικές λίστες και αρχικοποιεί τα δεδομένα τους σε 0. Αυτές οι 3 λίστες θα είναι παράλληλες μεταξύ τους αλλά και με τον πίνακα των μοναδικών ομάδων, και θα κρατούν τους πόντους και τα γκολ υπέρ/κατά για κάθε ομάδα
Οι λίστες δημιουργούνται με τη βοήθεια τις εντολής `declare -a` όπου δηλώνει

ουσιαστικά έναν πίνακα και τοποθετεί με χρήση μιας "for loop" μηδενικά παντού

- Αφού κάνει τις λίστες ξεκινάει να διασχίζει τον πίνακα με τους αγώνες και αφού είναι παράλληλος με τον πίνακα αποτελεσμάτων των αγώνων παίρνει το σκορ κάθε αγώνα. Πάλι αλλάζοντας το IFS σε "-" χωρίζει τις ομάδες κάθε αγώνα σε ht, at (home team, away team), και το αποτέλεσμα σε hs, as (home score, away score).
- Αφου λοιπών έχουμε τα γκολ κάθε ομάδας διασχίζουμε τη λίστα με τις μοναδικές ομάδες, και αφού βρούμε τις αντίστοιχες ομάδες τοποθετούμε τα γκολ που σκόραρε η κάθε ομάδα στις κατάλληλες θέσεις του πίνακα βάσει τη θέση της ομάδας στον πίνακα με τις ομάδες για τηρήσουμε την παραλληλότητα μεταξύ τους
Λχ: η ομάδα η οποία βρίσκεται στη 4η θέση του πίνακα με τις ομάδες, οι βαθμοί της και τα γκολ υπέρ/κατά θα βρίσκονταν και αυτά στις 4ες θέσεις των αντίστοιχων τους πινάκων
- Όταν τοποθετηθούν τα γκολ τότε σειρά έχουν οι βαθμοί του νικητή. Συγκρίνονται λοιπόν τα γκολ των 2 ομάδων και καθορίζεται ο νικητής. Ξέροντας τον νικητή, με την ίδια λογική με παραπάνω αναζητούμε τη νικήτρια ομάδα και προσθέτουμε +3 πόντους στην κατάλληλη θέση. Αν είναι ισοπαλία τότε αναζητούμε και τις 2 ομάδες και προσθέτουμε +1 στις θέσεις των πόντων τους
- Τέλος, πρέπει να παρουσιάσουμε την κατάταξη. Έτσι εκχωρούμε με την κατάλληλη στοίχιση τα αποτελέσματα σε ένα αρχείο και ακολούθως ταξινομούμε της γραμμές με βάση τους πόντους και μετά αλφαβητικά.
- Διαβάζουμε το αρχείο και το τυπώνουμε με κατάλληλη αρίθμηση και επαναφέρουμε το IFS.