

1ή Εργαστηριακή Άσκηση

Bash Scripts

Ομάδα: LAB21142517

| |
|-----------------------------|
| Μανώλης Πετράκος 2014030009 |
|-----------------------------|

| |
|-----------------------------------|
| Δημήτρης Καραμπάσογλου 2014030132 |
|-----------------------------------|

Η εργασία υλοποιήθηκε από κοινού.

Θέμα 1. Πρόγραμμα regr

Το πρόγραμμα δέχεται ως παραμέτρους εισόδου μία λίστα από αρχεία, τα οποία επεξεργάζονται ξεχωριστά και ανεξάρτητα. Απαιτείται τουλάχιστον ένα αρχείο για να λειτουργήσει. Κάθε αρχείο διαβάζεται ανά γραμμή και βάση του format που έχει δοθεί (num1:num2) δημιουργούνται οι πινάκες X και Y. Έπειτα υπολογίζονται τα αθροίσματα x, y, xy, x^2 χρησιμοποιώντας το εργαλείο awk γιατί η bash δεν υποστηρίζει πράξεις με floating point αριθμούς. Επίσης, ελέγχεται αν το διάνυσμα X είναι σταθερό (αν όλες οι τιμές του είναι ίδιες) και μετρείται το μέγεθος τους.

Το επόμενο βήμα είναι ο υπολογισμός των παραμέτρων a, b, c, err, για το οποίο χρησιμοποιείται πάλι το εργαλείο awk. Αυτό γίνεται βάση των τύπων που έχουν δοθεί στην εκφώνηση. Όμως, στην περίπτωση όπου το διάνυσμα X είναι σταθερό, δεν ισχύουν λόγο του ότι γίνεται διαίρεση με το μηδέν. Τότε, αντί του τύπου της προσέγγισης 'cY=aX+b' χρησιμοποιείται ο 'aX = cY + b'. Το a έχει τιμή 1 όπως το c στην προηγούμενη περίπτωση. Ο υπολογισμός των υπόλοιπων παραμέτρων γίνεται βάση των τύπων:

$$c = \frac{length * sum_xy - sum_y * sum_x}{length * sum_y^2 - sum_y * sum_y} \quad b = \frac{sum_x - c * sum_y}{length} \quad err = \sum_{i=0}^{length-1} (X[i] - (cY[i] + b))^2$$

Όμως, αυτοί οι τύποι είναι περιττοί, καθώς το διάνυσμα X είναι σταθερό, και τα αποτελέσματα είναι προβλέψιμα. Το c και το err έχουν πάντα την τιμή 0 ενώ το b είναι ίσο με το X. Για αυτόν τον λόγο, οι τύποι δεν υλοποιούνται στον κώδικα και τα αποτελέσματα εκχωρούνται απευθείας.

Απομένει η εμφάνιση των αποτελεσμάτων, όπου πρέπει να φαίνονται μέχρι δύο δεκαδικά ψηφία και να μην υπάρχουν trailing zeroes. Για αυτό χρησιμοποιείται το παρακάτω κομμάτι κώδικα.

```
b=`LC_ALL=C printf "%.2f" $b`  
echo $b | awk ' { sub("0+$",""); sub("\\.$",""); print} ' | tr '\n' ' '
```

Η printf στρογγυλοποιεί το αποτέλεσμα ώστε να έχει δύο δεκαδικά ψηφία. Αν το λειτουργικό σύστημα είναι στα Ελληνικά η έξοδος της θα έχει “,” αντί “.” και δεν θα λειτουργήσει το υπόλοιπο πρόγραμμα σωστά. Για αυτό αλλάζεται το locale με το ‘LC_ALL=C’ και εγγυάται η λειτουργία του προγράμματος

ανεξαρτήτου γλώσσας. Η printf θα αφήσει τα trailing zeroes αν υπάρχουν, για αυτό χρησιμοποιείται το awk και η εντολή sub. Η πρώτη sub αφαιρεί τα παραπάνω μηδενικά στο τέλος του αποτελέσματος και αν αφαιρεθούν δύο, τότε η δεύτερη θα αφαιρέσει το “.”. Τέλος, επειδή η print του awk προσθέτει ένα “\n” στο τέλος του string, χρησιμοποιείται η tr ώστε να κρατήσουμε το format της εκφώνησης.

Η λειτουργία του προγράμματος έχει ελεγχθεί με τα παρακάτω τεστ και τα αποτελέσματα έχουν διασταυρωθεί με εργαλεία του διαδικτύου.

| Αρχείο | 1.1 | 1.2 | 1.3 | 1.4 | 1.5 |
|----------|-----------|-----------|-----------|-----------|------------|
| Δεδομένα | 1.00:2.00 | 1.00:1.00 | 1.00:2.00 | 3.00:4.00 | 1.103:2.00 |
| | 1.00:1.00 | 1.00:1.00 | 4.00:2.00 | 4.00:5.43 | 1.103:1.00 |
| | 2.00:3.00 | 2.00:2.00 | 4.00:1.00 | 5.12:6.00 | 1.103:3.00 |
| | | | 2.00:5.00 | 2.00:8.00 | |
| | | | | 9.00:8.00 | |

Παράγονται τα παρακάτω αποτελέσματα:

```
[epetrakos@tower-pc project-1]$ bash regr 1.1 1.2 1.3 1.4 1.5
FILE: 1.1, a=1.5 b=0 c=1 err=0.5
FILE: 1.2, a=1 b=0 c=1 err=0
FILE: 1.3, a=-0.52 b=3.93 c=1 err=7.19
FILE: 1.4, a=0.24 b=5.16 c=1 err=10.19
FILE: 1.5, a=1 b=1.1 c=0 err=0
```

Παρατηρείται ότι το πρόγραμμα υποστηρίζει διαφορετικού μεγέθους αρχεία, σταθερά διανύσματα X (1.5) και το format της εξόδου είναι το επιθυμητό(δεν υπάρχουν παραπάνω “0” και “.” από όσα χρειάζονται).

Πηγές για την περίπτωση σταθερών διανυσμάτων X:

https://stats.stackexchange.com/questions/57685/line-of-best-fit-linear-regression-over-vertical-line?fbclid=IwAR34uyR8im3y3u1VmbpgGcLc7tWvo3Fb7F_LXrOXG0PY5T9Hku6u15JvFoA

Θέμα 2. Πρόγραμμα results

Το πρόγραμμα δέχεται ως είσοδο ένα αρχείο που περιέχει αγώνες και τα αποτελέσματά τους. Το πρώτο βήμα είναι να δημιουργηθεί ένας πίνακας με τις ομάδες που συμμετέχουν. Αυτό επιτυγχάνεται με την παρακάτω εντολή:

```
# briskei ola ta monadika onomata kai ta bazei se array
# read: -r = raw data, -d = delimiter, -a apoðikeush se pinaka
# grep: -o = krataei mono tis lekseis pou teriazoun, -E = gia na doulepsi me regular expresion
# <<< giati h read/readarray/mapfile gia na doulepsoun me pipe 8eloun while loop kai ginetai poluploko
# mapfile: -t = bgazei ta newline
mapfile -t array <<< $(cut -d ':' -f1 "${1}" | grep -o -E '^[^-]+' | sort | uniq)
```

Πρώτα, η cut κρατάει ό,τι υπάρχει πριν το “:”, δηλαδή κόβει τα αποτελέσματα. Έπειτα, η grep κόβει την διαχωριστική παύλα “-” ανάμεσα στις ομάδες και μένουν τα ονόματα των ομάδων. Επειδή μπορούν να εμφανιστούν πολλές φορές οι ίδιες ομάδες χρησιμοποιούνται οι sort και uniq ώστε να μείνει μια φορά η κάθε μια. Τέλος, η mapfile εκχωρεί τα αποτελέσματα σε ένα πίνακα. Στα σχόλια φαίνεται η λειτουργία του κάθε flag.

Ύστερα, γίνεται ο υπολογισμός των βαθμολογιών και των συνολικών goal. Το πρόγραμμα ελέγχει σε κάθε γραμμή αν μια ομάδα έπαιξε εντός ή εκτός και κάνει τους απαραίτητους υπολογισμούς. Αυτή η διαδικασία συνεχίζεται μέχρι να ελεγχθούν όλες οι ομάδες με όλες τις γραμμές. Σαν αποτέλεσμα, υπάρχουν τέσσερις πίνακες: τα ονόματα των ομάδων, οι βαθμολογίες τους, τα γκολ που έβαλαν και τα γκολ που δέχτηκαν.

Το τελευταίο βήμα είναι η ταξινόμηση των αποτελεσμάτων και η εμφάνισή τους. Αυτό γίνεται βάση της βαθμολογίας και σε περίπτωση ισοβαθμίας συγκρίνονται αλφαριθμητικά τα ονόματα των ομάδων. Η ταξινόμηση γίνεται με τον αλγόριθμο του bubble sort και τα αποτελέσματα εμφανίζονται με tabs όπως ζητείται.

Για τον έλεγχο της ορθής λειτουργίας του προγράμματος χρησιμοποιείται το παρακάτω τεστ και εμφανίζονται τα αποτελέσματα που ακολουθούν. Φαίνεται ότι ενώ οι Chelsea και Manchester city ισοβαθμούν, η Chelsea προηγείται στην κατάταξη καθώς το όνομα της είναι μικρότερο ως string. Τέλος, οι στήλες χωρίζονται με ένα TAB όπως και ζητείται, με αποτέλεσμα να μην υπάρχει στοίχιση.

Liverpool-Manchester united:0-4
Chelsea-Tottenham:2-2
Arsenal-Manchester city:1-3
Arsenal-Manchester united:2-5
Liverpool-Tottenham:1-1
Chelsea-Manchester city:4-3
Chelsea-Manchester united:0-1
Arsenal-Tottenham:2-3
Manchester city-Liverpool:0-0

```
[epetrakos@tower-pc project-1]$ bash results test0mades
1.      Manchester united      9      10-2
2.      Tottenham      5      6-5
3.      Chelsea 4      6-6
4.      Manchester city 4      6-5
5.      Liverpool      2      1-5
6.      Arsenal 0      5-11
```