Le meilleur défi de Friends of SAS

(probablement)

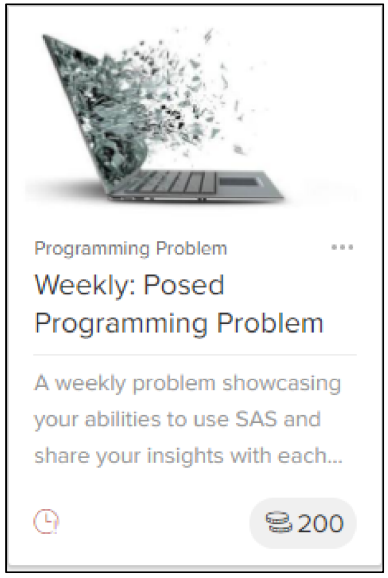
Par Mathieu Gaouette

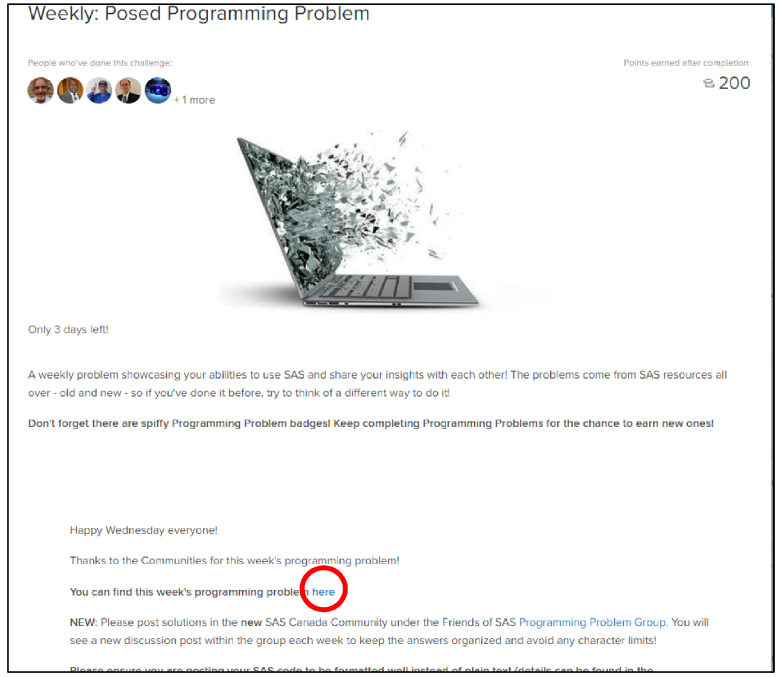
Pro spective MG

Plan

- Le problème
- Solution avec vecteurs
- Solution avec table hash
- Conclusion

Le problème


Programming Problem
Weekly: Posed Programming Problem
A weekly problem showcasing your abilities to use SAS and share your insights with each...
200


Weekly: Posed Programming Problem
People who've done this challenge:
+ 1 more
Points earned after completion:
200
Only 3 days left!
A weekly problem showcasing your abilities to use SAS and share your insights with each other! The problems come from SAS resources all over - old and new - so if you've done it before, try to think of a different way to do it!
Don't forget there are spiffy Programming Problem badges! Keep completing Programming Problems for the chance to earn new ones!
Happy Wednesday everyone!
Thanks to the Communities for this week's programming problem!
You can find this week's programming problem here.
NEW: Please post solutions in the new SAS Canada Community under the Friends of SAS Programming Problem Group. You will see a new discussion post within the group each week to keep the answers organized and avoid any character limits!

Le problème

Clinic Visits

Suppose you want to calculate one patient's 'maximum visit' of clinic, which changes over time.

Below is the raw data that is medical billing data. 'id 1' visited 3 different clinics and 'id 2' visits 2 different clinics.

In'20030124'(third row) 'id 1''s most visiting clinic is '12135' and maximum value is '2'.

However, in '20030607'(6th row), it is '75423' and maximum value is '3'. and in '20030815'(7th row) most visiting clinic is '12135' and '75423'. maximum value is still '3'.

'id 2''s most visiting clinic is '25875', and maximum value changes over time. it is 1, 2, 2, 3, 4, 5 in order.

Source

id	date	clinic
1	20020408	12135
1	20020715	12135
1	20030124	15109
1	20030308	75423
1	20030505	75423
1	20030607	75423
1	20030815	12135
2	20020615	25875
2	20020717	25875
2	20020824	55415
2	20020907	25875
2	20021215	25875
2	20030108	25875

Destination

id	date	clinic	count	max
1	20020408	12135	**1**	1
1	20020715	12135	**2**	2
1	20030124	15109	1	2
1	20030308	75423	1	2
1	20030505	75423	**2**	2
1	20030607	75423	**3**	3
1	20030815	12135	**3**	3
2	20020615	25875	**1**	1
2	20020717	25875	**2**	2
2	20020824	55415	**1**	2
2	20020907	25875	**3**	3
2	20021215	25875	**4**	4
2	20030108	25875	**5**	5

Count: compte du nombre de visite à cette clinique
Max: Nombre maximum de visite à une Clinique pour ce client

In addition, because it is national billing data, the type of clinic is more than 10 thousand, so it is impossible to make virtual array by clinic. SAS can't comprehend that much column.

Le problème

Source

id	date	clinic
1	20020408	12135
1	20020715	12135
1	20030124	15109
1	20030308	75423
1	20030505	75423
1	20030607	75423
1	20030815	12135
2	20020615	25875
2	20020717	25875
2	20020824	55415
2	20020907	25875
2	20021215	25875
2	20030108	25875

Analyse de la source

- Champ « id » identifiant le patient
- Champ « date » identifiant la date de visite
- Champ « clinic » identifiant la clinique

Ordre des données

- Trié par id et date

Solution avec vecteurs

```
data destination ;
    set source ;
    by id ;
    length count 8 max 8 ;
    retain Max .
           _clin1-_clin9999 . ;
    drop _: ;

    /* Création vecteurs */
    array cliniques (*) _clin1-_clin9999 ;

    /* On vide le vecteur pour un nouveau patient */
    if first.id then do ;
        do _i=1 to 9999;
           cliniques(_i) = 0 ;
        end ;
        max = 0 ;
    end ;

    /* Incrémentation du nombre de visite */
    cliniques(clinic) = cliniques(clinic) + 1 ;
    count = cliniques(clinic) ;

    max = max(max,count) ;
run ;
```

Identification source, ordre source, variables agrégées et « feuille de notes »

Initialisation « feuille de notes » pour nouveau patient.

Incrémentation visite clinique

Mise à jour maximum visites pour une clinique

Solution avec table hash

```
data destination ;
    set source ;
    by id ;
    length count 8 max 8 ;
    retain max . ;

    /* Création de la table hash */
    if _n_ = 1 then do ;
        declare hash h() ;
        h.defineKey('clinic');
        h.defineData('count');
        h.defineDone();
    end ;

    /* On vide la table hash pour un nouveau patient */
    if first.id then do ;
        h.CLEAR() ;
        max = 0 ;
    end ;

    /* Incrémentation du nombre de visite */
    if h.find() EQ 0 then do ;
        count = count + 1 ;
    end ;
    else do ;
        count = 1 ;
    end ;
    h.replace() ;

    max = max(max,count) ;
run ;
```

Identification source, ordre source, variables agrégées et « feuille de notes »

Initialisation « feuille de notes » pour nouveau patient.

Incrémentation visite clinique

Mise à jour maximum visites pour une clinique

Conclusion

Table hash

- Très performant
- Utilisation optimale de la mémoire
- Syntaxe peu intuitive

Vecteurs

- Syntaxe intuitive
- Performant (généralement)
- Peu optimale dans certains cas (comme le notre)