kadaoui_alexandre_Manipulation_des_facteurs

Alexandre

23/12/2020

R Markdown

Dans le cadre de notre partiel, nous devons réaliser un total de 12 travaux retracant notre parcours et notre travail durant les 30 heures de cours.
Le travail à faire est le suivant :
- Une entête comportant un titre, un lien Github avec le ou les noms des auteurs.
- Une synthese de ce travail
- Un extrait commenté avec des parties de codes clé avec explication et commentaire.
- Une évalutation du travail avec nos 5 criteres.
- Une conclusion du travail

Definition des 5 critères de notations :

1) Effort de présentation :
2) Le knit est réalisable et bien présenté.
3) Explications simples et efficaces.
4) Le Code reproductible à d'autres DataFrame avec facilité.
5) Description des fonctions utilsés et du raisonnement.

Manipulation des facteurs

Travail réalisé par " Claire Mazzucato " le 09/11/2020.

GitHub: https://github.com/clairemazzucato/PSBX/tree/main/Manipulation%20des%20facteurs (https://github.com/clairemazzucato/PSBX/tree/main/Manipulation%20des%20facteurs)

Synthese :

Les facteurs sont une forme particulière de vecteur destinés à faciliter la manipulation de données qualitatives en stockant les éléments mais également l'ensemble des modalités différentes possibles (les autres options inutilisées en somme) dans un attribut accessible via la commande **levels**
Ces facteurs peuvent être non ordonnés (homme/femme, pays etc…) ou ordonnés (niveaux de ski, classement etc…)

Extrait commenté du code :

Création de facteurs

1 - La fonction **factor()** permet de créer un facteur en définissant directement ses différents éléments

```
sexe <- factor(c("H", "H", "F", "H", "H", "F", "F", "F"))
sexe
```

```
## [1] H H F H H F F F
## Levels: F H
```

2 - La fonction **as.factor** permet de convertir un vecteur en facteur

```
salto <- c(1:5,5:1) #création d'un vecteur classique
salto
```

```
##  [1] 1 2 3 4 5 5 4 3 2 1
```

```
salto.f <- as.factor(salto) #transofmration du vecteur en facteur
salto.f
```

```
##  [1] 1 2 3 4 5 5 4 3 2 1
## Levels: 1 2 3 4 5
```

3- La fonction **ordered** permet de créer des facteurs ordonnés grâce au terme **levels**

```
niveau <- ordered(c("débutant","débutant","champion",
                    "champion","moyen","moyen","moyen",
                    "champion"),
levels=c("débutant","moyen","champion"))
niveau
```

```
## [1] débutant débutant champion champion moyen    moyen    moyen    champion
## Levels: débutant < moyen < champion
```

Niveaux d'un facteur

Pour connaitre les niveaux d'un facteur on utilise la fonction **level**

```
levels(sexe)
```

```
## [1] "F" "H"
```

Par exemple ici, on établie les niveaux d'un dé à 6 faces dans le terme **levels**

```
de <- factor(c(3, 2, 2, 1, 3, 1, 3, 1), levels = c(1, 2, 3, 4, 5, 6))
de
```

```
## [1] 3 2 2 1 3 1 3 1
## Levels: 1 2 3 4 5 6
```

```
str(de) #afficher la structure du facteur et ses attributs ordonnés
```

```
##  Factor w/ 6 levels "1","2","3","4",..: 3 2 2 1 3 1 3 1
```

Modifier les valeurs d'un facteur

On modifie ici les valeurs en position 2 et 3 par des valeurs 5 et 6

```
de[c(2, 3)] <- c(6, 5)
de
```

```
## [1] 3 6 5 1 3 1 3 1
## Levels: 1 2 3 4 5 6
```

On peut modifier l'ordre des niveaux d'un facteur existant en utilisant l'option **levels**

```
sexe <- factor(sexe, levels = c("H", "F"))
sexe
```

```
## [1] H H F H H F F F
## Levels: H F
```

Evaluation du travail :

1) Effort de présentation : Le RMD ainsi que le PDF sont bien présentés
2) Le knit est réalisable et bien présenté : Le Knit ne pose aucun soucis, les fonctinos s'exécutent bien
3) Explications simples et efficaces : Les explicatoin sont simples et faciles à comprendre tout en explorant un panel varié et complets de manipulations possibles et concrètes
4) Le Code reproductible à d'autres DataFrame avec facilité : Grace à la qualité des explication, le code est facilement reproductible à d'autres dataframe
5) Description des fonctions utilsés et du raisonnement : les principales fonctions utilisées ici ont été les fonctions **factor()**, **as.factor** et **ordered** afin de créer les facteurs ainsi que **level** afin gérer les niveaux au sein de ceux-ci.

Conclusion :

En conclusion, ce tutoriel présente de manière simple, complète et succinte le concept de facteurs et leur gestion des divers niveaux afin de stocker et manipuler des données qualitatives sous forme de vecteurs.