kadaoui_alexandre_Manipulation_Strings

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 de Strings

Travail réalisé par " Léonard Boisson " le 19/11/2020.

GitHub: "https://github.com/LeoBsn/PSB-PROJECT (https://github.com/LeoBsn/PSB-PROJECT)"

Synthese :

La manipulation de strings (chaines de caractères attachés entre eux) est possible en R via de multiples fonctions que ce tuto présente.
La librairie **stringr** nous offre également des fonctions permettant le traitement et la manipulation de strings
Ces fonctions permettent également le nettoyage des données brutes afin de ne récupérer que les données clefs du document et les séparer du "bruit"

Extrait commenté du code :

```
#install.packages('stringr')
library(stringr)
library(readxl)
```

Fichier excel de l'exemple manquant dans le github

Dans l'exemple suivant, on stock dans un objet "d" le contenu d'un fichier excel en attribuant à chacune de ces trois colonnes une valeur de "text" grâce à la fonction **read_excel**

```
d <- read_excel("exemple.xlsx", col_types = c("text", "text", "text"))
```

###Concaténer des chaînes de caractères avec **str_c**, ici on concatène les colonnes 1 et 2 du fichier excel contenu dans d, en reliant les deux termes par un " - ".

```
str_c(d$colonne1, d$colonne2, sep = " - ")
```

Conversion de la colonne 1 en minuscules et de la colonne 2 en majuscules

```
str_to_lower(d$colonne1)
str_to_upper(d$colonne2)
```

On passe en majuscule la première lettre de chaque mot de la colonne 3

```
str_to_title(d$colonne)
```

Trouver la longueur d'une chaîne de caractère avec **str_lenght**: renvoie le nomnre de caractères dans chaque élement de la colonne 4

```
str_length(d$colonne4)
```

La fonction **str_trim** nous permet de supprimer les espaces au début et à la fin de la chaîne de caractère

```
str_trim('       Le vent souffle sur les plaines de la Bretagne armoricaine       ')
```

```
## [1] "Le vent souffle sur les plaines de la Bretagne armoricaine"
```

La fonction **str_split** nous permet ici de découper cette chaîne en fonction d'un caractère précisé, ici un espace

```
str_split('Et un et deux et trois zéro', ' ')
```

```
## [[1]]
## [1] "Et"    "un"    "et"    "deux"  "et"    "trois" "zéro"
```

str_detect peut être utilisé afin de détecter la présence d'un craactère ou d'une chaîne de craactère, ici, on souhaite détecter la présence du mot "bonjour" dans la colonne 2

```
str_detect(d$colonne2, 'bonjour')
```

str_count renvoie le nombre de fois ou le caractère choisis est présent, ici le caractère "o" dans la colonne 2

```
str_count(d$colonne2, 'o')
```

str_subset renvoie seulement les éléments dans laquelle la chaîne de caractère choisie est présente, ici le terme bonjour dans la colonne 2

```
str_subset(d$colonne2, 'bonjour')
```

Evaluation du travail :

1) Effort de présentation : Le PDF ainsi que le RMD sont bien présentés et structurés
2) Le knit est réalisable et bien présenté : Le knit est bien réalisable à condition de ne pas exécuter les chunks de code nécessitant un fichier excel sur lequel s'appuyer.
3) Explications simples et efficaces : Les explications concernant les concepts, le fonctionnement et le code sont claires, succintes et simples. Il est très facile d'intégrer les concepts et le fonctionnement du code.
4) Le Code reproductible à d'autres DataFrame avec facilité : La qualité d'explication du code ainsi que sa sub division efficace (le rendant très digeste) permettent de bien s'approprier le code et son fonctionnement et ainsi d'être facilement en mesure de le reproduire afin d'analyser tout type de fichier
5) Description des fonctions utilsés et du raisonnement : Les fonctions utilisées ont des fonctionnement somme toute très similaire à ce que 'lon pourrait trouver sur excel, le fonctionnement est donc assez familier, parmi les fonctions clefs on trouve ici **read_excel**, **str_c**, **str_trim**, **str_subset** etc…

Conclusion :

En conclusion ce RMD illustre de manière simple et efficace la gestion de strings via R, cela au moyen de fonctions dont la logique rappelle beaucoup Excel.