**Calcul de complexité**

**Explication :**

**Paramètres d'entrée :**

**las :** L'objet LAS en entrée représentant le nuage de points.

**random\_value :** Une valeur aléatoire entre 0 et 1 qui détermine la proportion de points à conserver.

**proportion <- random\_value :** La valeur aléatoire sert de proportion de points à conserver. Cela signifie qu'une fraction aléatoire de points sera conservée du nuage de points original.

**n\_points <- ceiling(npoints(las) \* proportion) :** Calculer le nombre de points à conserver en multipliant le nombre total de points dans l'objet LAS original (npoints(las)) par la proportion spécifiée. La fonction ceiling est utilisée pour garantir qu'au moins un point est conservé.

Échantillonnage aléatoire des points :

**indices\_échantillonnés <- sample(1:npoints(las), size = n\_points, replace = FALSE) :** Sélectionner de manière aléatoire les indices des points à conserver à partir de l'objet LAS original. Cette étape garantit que les points sélectionnés sont répartis dans l'ensemble du nuage de points d'origine.

**Extraction des points échantillonnés :**

**las\_épuré <- las[indices\_échantillonnés] :** Extraire les points échantillonnés de l'objet LAS original pour créer un nouvel objet LAS représentant le nuage de points épuré.

**Retour de l'objet LAS épuré :**

**return(las\_épuré) :** La fonction retourne l'objet LAS épuré, qui contient un sous-ensemble de points choisis de manière aléatoire à partir du nuage de points d'origine.

Cette fonction offre essentiellement un mécanisme de base pour réduire la densité d'un nuage de points en fonction d'une valeur aléatoire, mais il est important de noter que cette approche peut ne pas convenir à toutes les situations, et le package lidar propose des méthodes plus avancées pour le traitement des nuages de points.

**La complexité de la méthode :**

La complexité totale de la fonction est dominée par l'étape d'échantillonnage aléatoire, donc la complexité totale serait

O(n points). Cela signifie que le temps d'exécution de la fonction augmente linéairement avec le nombre total de points dans l'objet LAS.

Il est important de noter que la complexité exprimée ici est basée sur une analyse théorique. Dans la pratique, les performances réelles peuvent être influencées par des facteurs tels que la mise en œuvre spécifique, la machine sur laquelle le code est exécuté, etc.