

Travaux Dirigés ISV51 - Matrices, Listes, Tableaux de données

Julien Chiquet

2 octobre 2015

Objectifs de la séance

- création/génération/manipulation élémentaires de matrices, listes et tableaux de données
- résumés numérique et représentations graphiques élémentaires

Exercice 1: opérations algébriques élémentaires

1. Créer deux vecteurs u et v de taille 6. Calculer $u^T v$, uv^T avec les commandes `%*%` puis les commandes `crossprod` et `tcrossprod`.
2. Créer une matrice A à 3 lignes et 2 colonnes à partir du vecteur u puis une matrice B à 2 lignes et 3 colonnes à partir du vecteur v . Effectuer un produit matriciel entre A et B puis entre B et A .
3. À partir des matrices A et B , calculer le produit scalaire entre les deux dernières lignes de A et la 1ère colonne de B .
4. Concaténer les matrices A et B^T en colonne pour former une matrice 6×4 . De même, concaténer A^T et B en ligne pour obtenir une matrice 4×6 .

Exercice 2: création et manipulation de matrices

1. Données Iris

- a) Charger les valeurs numériques des données iris à l'aide de la commande
- b) Donner la dimension de la matrice ainsi construite. Trouver la plus grande valeur observée. Donner le numéro de ligne et de colonne correspondant.
- c) Calculer la moyenne en ligne et en colonne, d'abord avec les commandes `rowSums`, `colSums` et `nrow`, `ncol`, puis à l'aide de la commande `apply`. Quel individu a la plus grande longueur de Sépale ? Largeur de Pétale
- d) Représenter le graphe des paires de variable à l'aide de la commande `pairs`.

2. Données microarray

- a) Charger les valeurs numériques des données de puces pour le cancer à l'aide de la commande

```
microarray <- as.matrix(read.table("http://statweb.stanford.edu/~tibs/ElemStatLearn/datasets/14cancer.x
```

- b) Calculer la covariance entre les échantillons. Représenter le résultat sous forme d'image. Transformer la covariance en corrélation et représenter à nouveau cette image. Représenter ensuite le résultat de la fonction `heatmap`.

Exercice 3: création et manipulation de listes

On utilise un programme permettant de calculer le nombre d'occurrence des 4 nucléotides "a", "c", "g" et "t" dans une séquence d'ADN. Celui-ci renvoie une liste comportant 4 éléments, chacun étant un vecteurs décrivant les indices des occurrences des lettres correspondantes.

- Considérons la séquence "AATTCCTCCCGTGACGAAATATA". Créer l'objet R correspondant à l'exécution du programme ci-dessus.
 - Déterminer le nombre d'occurrence de chaque lettre dans la séquence à partir de cette liste.
- On dispose maintenant de 3 chaînes "ATTTCG", "CCGT" et "GCGAGG". Créer une liste comprenant 3 entrées, chacune étant une liste comme celle décrite aux deux questions précédentes.
 - Déterminer la longueur de chaque séquence à partir de cette liste
 - Déterminer le nombre d'occurrence de chaque nucléotide dans chacune des listes. Renvoyer le résultat sous forme de matrice 3 x 4 (on pourra s'aider de la fonction `sapply`).

Exercice 4: création et manipulation de data.frame

- Charger le tableau de données `diamonds` (commande `data`). Vérifier qu'il s'agit bien d'un `data.frame`. Déterminer les noms des variables considérées et leur nature. Faites un résumé numérique.
- À l'aide de la commande `subset`, extraire les entrées du tableau telles que
 - les diamants soit de qualité `Premium`
 - le carat soit supérieur à 3
 - le volume (approximatif) soit supérieur à 500mm^3
 - la qualité soit idéal, le prix inférieur à 1000 et le carat supérieur à .5. Déterminer la répartition des couleurs pour ce sous-ensemble
- Déterminer le prix moyen par classe de qualité. Même question par intervalle de carat (vous créez une variable factorielle composée de 6 intervalles à l'aide la fonction `cut`).
- Tracer le volume en fonction du prix, le carat en fonction du prix. Représenter les boxplot de carat, prix et profondeur par classe de qualité et par couleur.
- Pour chaque triplet (`cut,color,clarity`), renvoyer le prix moyen.