

Formation à blanc

Initiation R

E.HIRCHAUD
F.SIMONET

BiRD

3 septembre 2013



Sommaire

- 1 Notions informatiques
 - Système d'exploitation
 - Les applications
 - Chemins et fichiers
 - Types de mémoires
 - Synthèse

- 2 R
 - Aperçu
 - Objectif du TP
 - Comparaison Excel R
 - Les objets



Les système d'exploitation : OS (Operating System)



debian



FreeBSD



solaris



Les applications : logiciels, programmes,

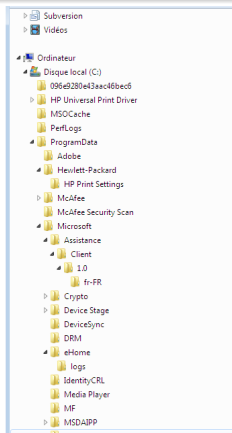


Arborescence

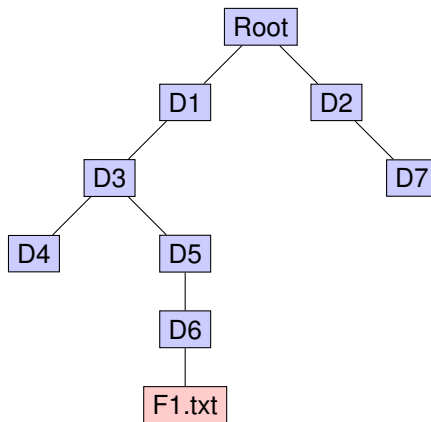
Apprélation

- Fichiers données numériques de type variés
- Directory = Répertoire = Dossier : contient des fichiers.

bin			- dossier
Documents			- dossier
Formation			- dossier
Formation_puce2012			- dossier
data			- dossier
15K_alea.atr	755 octets	inconnu	
15K_alea.cdt	3,0 Mo	inconnu	
15K_alea.gtr	716,4 ko	inconnu	
15K_alea.txt	3,8 Mo	document texte brut	
annotation_echantillons.txt	291 octets	document texte brut	
bg_matrix_lowess.txt	15,1 Mo	document texte brut	
matrix_lowess.txt	10,8 Mo	document texte brut	
Resultats			- dossier
filtrage_invariants			- dossier
filtrage_non_exprimes			- dossier
filtrage_non_exprimes_Excel			- dossier
matrix_totale.filtree.txt	5,7 Mo	document texte brut	
Scripts			- dossier
Feuille_route_TP_filtrage.doc	227,3 ko	Document Microsoft Word	
Formation_R_2013			- dossier
image			- dossier
logo.jpg	72,5 ko	image JPEG	
OS.png	66,4 ko	image PNG	



Schématisation



La navigation

Deux types de navigation

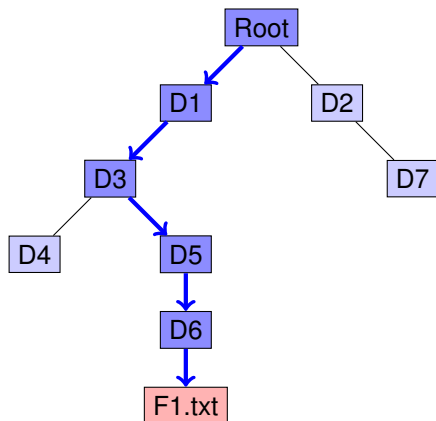
- Chemin absolu
- Chemin relatif

Symboles utilisés dans la navigation

- La racine : Windows une lettre, linux et mac, symbole /
- Séparateur de répertoire : Windows : \, linux et mac /
- Le Répertoire courant : . (point)
- Le Répertoire parent : .. (deux points)



Formalisation : Chemin absolu

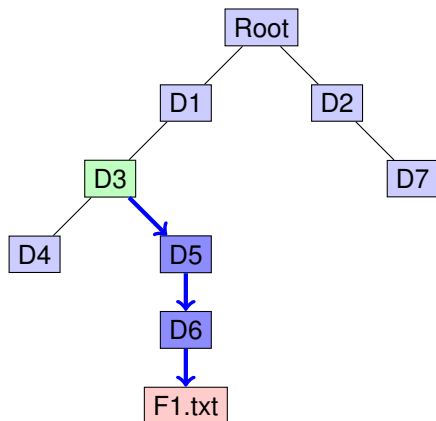


Chemin absolu

- A partir de la racine
- Root/D1/D3/D5/D6/F1.txt



Formalisation : Chemin relatif exemple 1

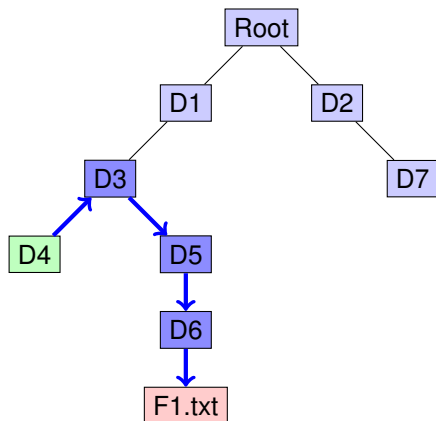


Chemin relatif

- D5/D6/F1.txt



Formalisation : Chemin relatif exemple 2



Chemin relatif

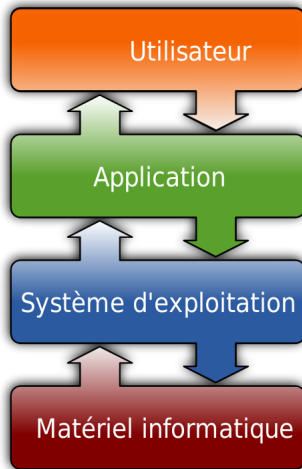
- ../D5/D6/F1.txt



Les mémoires



Synthèse



R : Apperçu

- Créée par Ross Ihaka et Robert Gentleman (1996)
- C'est un logiciel libre et gratuit
- Disponible sur les systèmes d'exploitation les plus utilisés
- Utilisé dans de nombreux domaines dont la bio analyse.



Objectif du TP

- **Assimiler le vocabulaire**
- Se servir de R comme d'une calculatrice
- Écrire et modifier des lignes de commande
- Utiliser un script déjà écrit
- Savoir où trouver de l'aide (documentation)
- Utiliser un éditeur convivial (RStudio)



Objectif du TP

- Assimiler le vocabulaire
- Se servir de R comme d'une calculatrice
- Écrire et modifier des lignes de commande
- Utiliser un script déjà écrit
- Savoir où trouver de l'aide (documentation)
- Utiliser un éditeur convivial (RStudio)



Objectif du TP

- Assimiler le vocabulaire
- Se servir de R comme d'une calculatrice
- Écrire et modifier des lignes de commande
- Utiliser un script déjà écrit
- Savoir où trouver de l'aide (documentation)
- Utiliser un éditeur convivial (RStudio)



Objectif du TP

- Assimiler le vocabulaire
- Se servir de R comme d'une calculatrice
- Écrire et modifier des lignes de commande
- Utiliser un script déjà écrit
- Savoir où trouver de l'aide (documentation)
- Utiliser un éditeur convivial (RStudio)



Objectif du TP

- Assimiler le vocabulaire
- Se servir de R comme d'une calculatrice
- Écrire et modifier des lignes de commande
- Utiliser un script déjà écrit
- Savoir où trouver de l'aide (documentation)
- Utiliser un éditeur convivial (RStudio)



Objectif du TP

- Assimiler le vocabulaire
- Se servir de R comme d'une calculatrice
- Écrire et modifier des lignes de commande
- Utiliser un script déjà écrit
- Savoir où trouver de l'aide (documentation)
- Utiliser un éditeur convivial (RStudio)



Comparaison Excel R

Excel

R



Comparaison Excel R

Excel	R
Cellule	Élément simple
Plage de données	data.frame matrix, list, vector



Comparaison Excel R

Excel	R
Cellule	Élément simple
Plage de données	data.frame matrix, list, vector
Valeur	Valeur (value)
Format	Type



Comparaison Excel R

Excel	R
Cellule	Élément simple
Plage de données	data.frame matrix, list, vector
Valeur	Valeur (value)
Format	Type
Fonction	Fonction
Macro	Script



Les Objets

Deux types d'objets à retenir

- Les objets de données
- Les fonctions



Les Objets de données

- Un nom : que l'on appelle variable
- Valeur(s)
- Les valeurs ont un type :
 - ▶ numérique : 1,2, 3.14
 - ▶ chaîne de caractères : A,B genes
 - ▶ logique : TRUE/FALSE

Les objets sont temporaires, ils sont stockés dans la mémoire vive de l'ordinateur. Il faudra donc les sauvegarder.



Catégories de données

- **vector** \Rightarrow vecteur (type homogène)
- matrix \Rightarrow matrice (type homogène)
- data.frame \Rightarrow tableau de données (type hétérogène)
- factor \Rightarrow classe de paramètres qualitatifs (type homogène)
- list \Rightarrow liste(type hétérogène)



Catégories de données

- `vector` \Rightarrow vecteur (type homogène)
- `matrix` \Rightarrow matrice (type homogène)
- `data.frame` \Rightarrow tableau de données (type hétérogène)
- `factor` \Rightarrow classe de paramètres qualitatifs (type homogène)
- `list` \Rightarrow liste(type hétérogène)



Catégories de données

- `vector` \Rightarrow vecteur (type homogène)
- `matrix` \Rightarrow matrice (type homogène)
- `data.frame` \Rightarrow tableau de données (type hétérogène)
- `factor` \Rightarrow classe de paramètres qualitatifs (type homogène)
- `list` \Rightarrow liste(type hétérogène)



Catégories de données

- `vector` \Rightarrow vecteur (type homogène)
- `matrix` \Rightarrow matrice (type homogène)
- `data.frame` \Rightarrow tableau de données (type hétérogène)
- `factor` \Rightarrow classe de paramètres qualitatifs (type homogène)
- `list` \Rightarrow liste(type hétérogène)



Catégories de données

- `vector` \Rightarrow vecteur (type homogène)
- `matrix` \Rightarrow matrice (type homogène)
- `data.frame` \Rightarrow tableau de données (type hétérogène)
- `factor` \Rightarrow classe de paramètres qualitatifs (type homogène)
- `list` \Rightarrow liste(type hétérogène)



Création d'objets

```
# Objet Simple
monNombre <- 4.5

# Vecteur numerique
monVecNum <- c(500, 452, 8)

# Vecteur de caractaire
monVecCara <- c("A", "B", "C")

# Matrice de 10 colones et 50 lignes
maMatrice <- matrix(rnorm(500), ncol=10)

# data frame formé deux vecteurs:
monDataFrame <- data.frame(monVecNum, monVecCara)
```



Notion d'indices index

- Les indices sont entouré de crochets
- Il s'agit de la position d'une valeur
- Pour les vecteurs, simple indice. $[i]$
- Pour les data.frame et matrice, double indice, $[Ligne, Colone]$



Vecteur

```
monVecNum <- c(500 , 452, 8)  
element3 <- monVecNum[3]
```

<i>Indices</i>	[1]	[2]	[3]
<i>Valeurs</i>	500	452	8



Matrices

```
maMatrice <- matrix(1:15, ncol=5)
maLigne2 <- maMatrice[2, ]
maColone3 <- maMatrice[, 3]
elementL2C3 <- maMatrice[2,3]
```

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	1	2	3	4	5
[2,]	6	7	8	9	10
[3,]	11	12	13	14	15



Data Frame

```
Ech <- c(1:6)
Type <- c("WT", "WT", "MUT1", "MUT1", "MUT2", "MUT2")
monDataFrame <- data.frame(Ech, Type)

#Reccupere la colonne Type
colType <- monDataFrame$Type
colType2 <- monDataFrame[, 2]
```

	Ech	Type
[1,]	1	WT
[2,]	2	WT
[3,]	3	MUT1
[4,]	4	MUT1
[5,]	5	MUT2
[6,]	6	MUT2



Les Fonctions

- On également un nom
- Une description (leur rôle)
- Des arguments, paramètres
- Retourne un ou des résultats
- Créent, modifient et informent sur les données
- Contiennent des arguments et des instructions



Utilisation d'une fonction

Une fonction c'est une recette de cuisine : La recette de la pâte à pizza

Le quatre quarts : ingrédients

- 500g de farine
- 250ml d'eau
- 20g de levure de boulanger fraîche



La pâte à pizza : une suite d'actions ordonnées

- 1 Verser la farine dans un saladier, y creuser un puits et ajouter l'eau
- 2 Dans un petit bol, faire fondre la levure dans un peu d'eau tiède.
- 3 ect. . .



Un résultat

Une pâte à pizza prête à garnir.



En informatique :

- La fonction à un nom : `preparerPateAPizza`
- Les ingrédients sont des arguments : `poidsFarine`, `volumeEau`, `poidsLevure`
- Les actions sont des suites d'instruction (ligne de commandes)
- Le retour de la fonction est un objet que l'on peut stocker dans une variable : `maPateAPizza`.

```
maPateAPizza <- preparerPateAPizza(poidsFarine = 250, volumeEau = 250,  
poidsLevure = 20)
```



Ce qu'il faut retenir sur les fonctions

- Les reconnaître
- Identifier les arguments
- Modifier les arguments
- Stocker le résultat dans une variable.
- Lire la documentation de la fonction pour connaître son utilisation
- Il n'est pas essentiel de connaître son fonctionnement dans le détail.



Utilisation des arguments

Généralité sur les arguments.

- L'ordre des arguments n'a pas d'importance si leur noms est écrits
- Les variables doivent être dans un ordre précis si les noms ne sont pas écrits
- Certains arguments ont une valeur par défaut, si on ne la change pas, aucun besoin de l'écrire.
- Les valeurs attribuée aux arguments peuvent être des variables.



Package

- Les packages contiennent un ensemble de fonctions et parfois de jeux de données.
- `library(ggplot2)`
- BioConductor : ensemble de package pour la bioanalyse



Script/Programme

- Un script permet l'exécution automatique d'une série d'instruction, de fonction...
- Ces instructions sont écrites dans un fichier texte qui a pour extension .R
- Un script comporte généralement des lignes de commentaire, précédées par le symbole #.

```
1 #Preparation de la pate a Pizza
2 maPateAPizza <- preparerPateAPizza (poidsFarine = 250, volumeEau = 250,
   poidsLevure = 20)
3
4 #Stockage des ingredients
5 mesIngredient <- c("Tomates", "Mozzarella", "Olives noires")
6
7 #Garnir la pizza
8 maPizzaGarni <- garnirPizza(pate = maPateAPizza, ingredients =
   mesIngredients)
9
10 #Cuir la pizza
11 pizzaCuite <- cuirePizza(pizzaGarni = maPizzaGarni)
```

Règles de nomenclatures

- Importance de la casse (majuscule/minuscule)
 - ▶ : pizza \neq Pizza
- Informatique anglo-saxone
 - ▶ Ne pas nommer les noms des objets avec des accents
 - ▶ Le point sert de décimal, la virgule non !
- Ne JAMAIS mettre d'espace dans un nom
- Ne JAMAIS commencer un nom par un chiffre
- Eviter d'utiliser des symboles (+ - / ...)
- Mots réservés



Objets



Objets

Fichiers,
script ...



Synthèse

RAM
(Temporaire)

Objets

Fichiers,
script ...



Synthèse

RAM
(Temporaire)

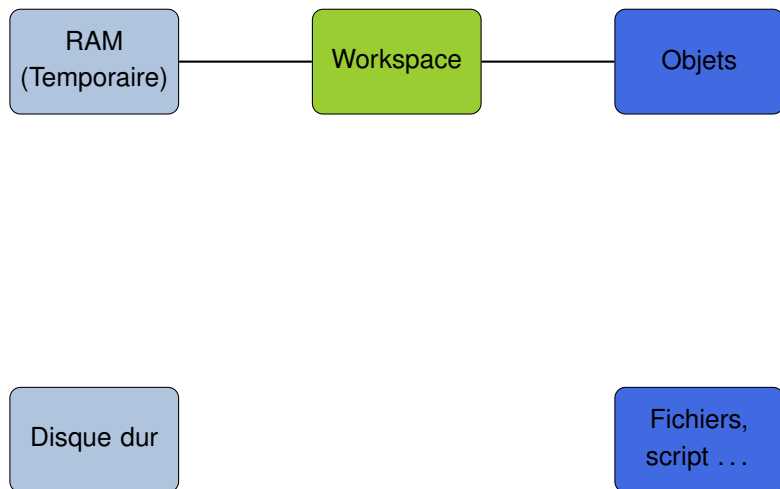
Objets

Disque dur

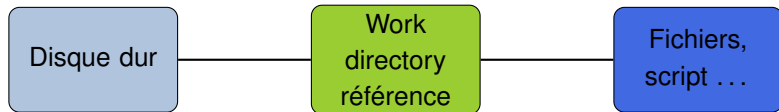
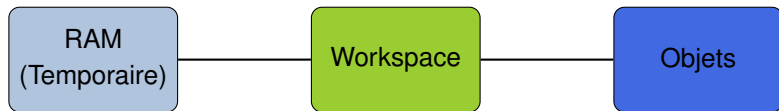
Fichiers,
script ...



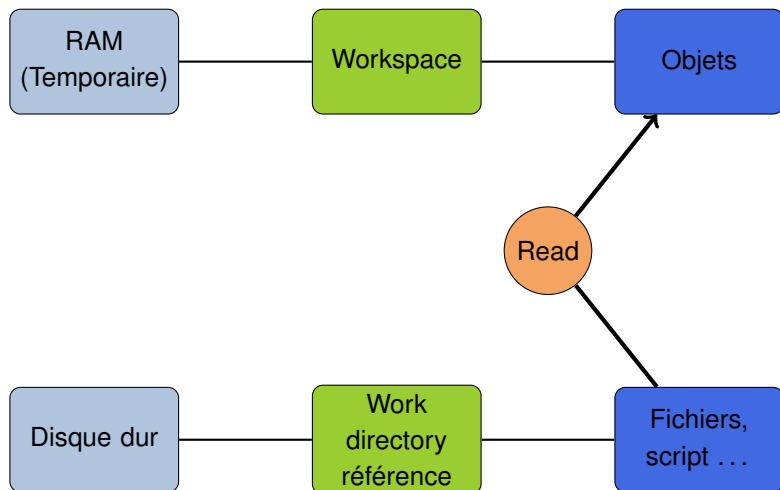
Synthèse



Synthèse



Synthèse



Synthèse

