

# Initiation R

E.Hirchaud

BiRD

3 septembre 2013



# Sommaire

## 1 Notions informatiques

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

## 2 R

- Apperçu
- Objectif du TP
- Comparaison Excel R



# Les système d'exploitation : OS (Operating System)



debian



FreeBSD



solaris



# Les applications : logiciels, programmes,

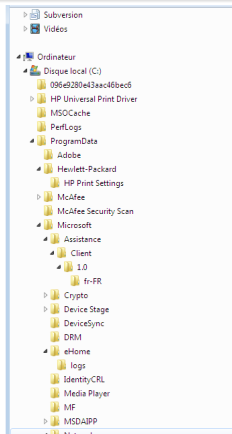


# Arborescence

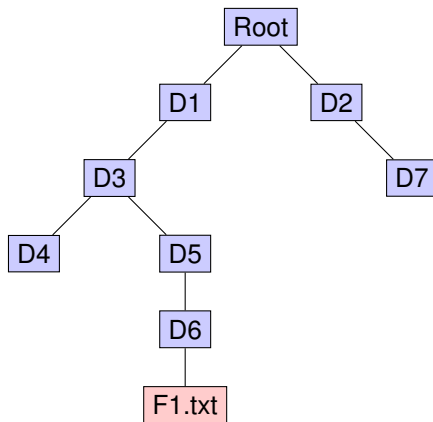
## Apprélation

- 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

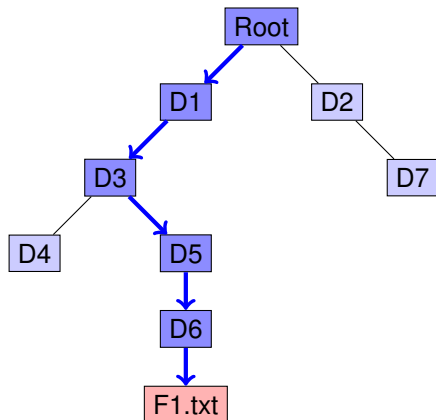
- Chemin absolu
- Chemin relatif

## 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



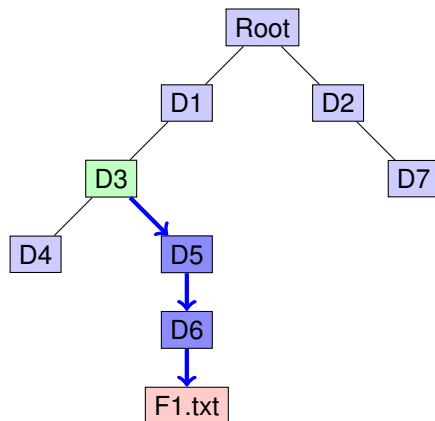
## Chemin absolu

- A partir de la racine (C:)
- Root/D1/D3/D5/D6/F1.txt





# Formalisation : Chemin relatif exemple 1

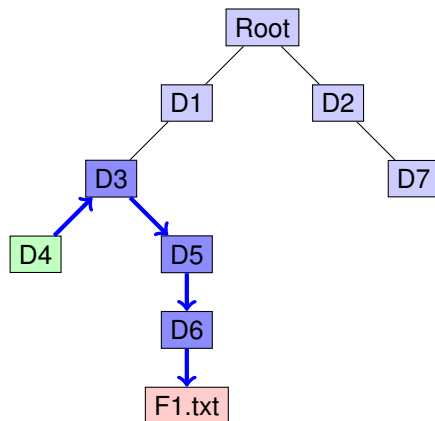


Chemin relatif

- D5/D6/F1.txt



## Formalisation : Chemin relatif exemple 2



Chemin relatif

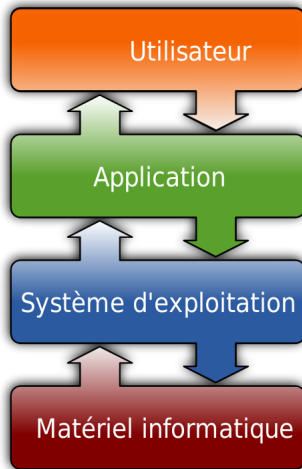
- ../D5/D6/F1.txt



# Les mémoires



# Synthèse



# R : Apperçu

- Créée par Ross Ihaka et Robert Gentleman (1996)
- C'est un logiciel libre et gratuit
- Il est basé sur le langage S qui est propriétaire
- 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
Cellule	Objet (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)



# Les Fonctions

- On également un nom
- Une description (leur rôle)
- Des arguments, paramètres
- Retourne un 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 du quatre quarts

## Le quatre quarts : ingredients

- 4 oeufs
- 200g de farine
- 200g de beurre
- 200g de sucre



## Le quatre quarts : une suite d'actions ordonnées

- 1 Préchauffez le four à 210C.
- 2 Mélangez bien le sucre et le beurre fondu.
- 3 Cassez les oeufs et réservez les blancs.
- 4 Ajoutez les jaunes d'oeufs dans la préparation à base de beurre un par un en mélangeant bien.
- 5 ...



## Un résultat

Le quatre quart prêt à manger



## En informatique :

- Le nom de la fonction suit certaines règles : `cuisinerUnQuatreQuarts`
- Les ingrédients sont des arguments : `nbOeufs`, `poidFarine`, `poidSucre`, `poidBeurre`
- Les actions sont des suites d'instruction (ligne de commandes)
- Le retour de la fonction est un objet que l'on peut stoker dans une variable : `monQuatreQuarts`.
- `monQuatreQuarts <- cuisinerUnQuatreQuarts(nbOeuf = 4, poidFarine = 200, poidBeurre = 200, poidSucre = 200)`





# Ce qu'il faut retenir sur les fonctions

- Les reconnaître
- Identifier les arguments
- Modifier les arguments
- Stocker le résultats 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.



# Règles de nomenclatures

- Importance de la casse (majuscule/minuscule)
  - ▶ : monQuatreQuarts  $\neq$  MonQuatreQuarts
- 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 (+ - / ...)

