

## Advanced R Programming : Work efficiently with R

Advanced R course teaches students more sophisticated R skills, including using advanced regular expressions, machine learning, random effects modeling, Bayesian Inference, advanced R time series, and much more.

### Détails

- **Code** : DB-LRA
- **Durée** : 3 jours ( 21 heures )

#### Public

- Architectes
- Consultants
- Consultants informatiques
- Développeurs
- Ingénieurs
- Professionnels de l'IT

#### Pré-requis

- R Programming experience or R Programming course

### Objectifs

- Use advanced regular expressions in R
- Apply advanced missing data techniques
- Work with advanced R time series
- Use `data.table` for big data
- Work with linear models

### Programme

#### Organiser son travail sous R

- Travailler en projet R : notion de working directory, workspace, history
- Architecture de son projet R : data, plots, images, scripts, etc..
- Bonnes pratiques pour la création de fichiers de données
- Importation et exportation de fichiers avec le package `here`
- Mettre à jour ses packages
- Mettre à jour R et R Studio

#### Manipuler facilement ses données avec le package `dplyr`

- Introduction au package `tidyverse` et à la notion de pipe
- Filtrer des lignes avec la fonction `filter()`
- Sélectionner des colonnes (variable) avec la fonction `select()`
- Création de nouvelles variables avec la fonction `mutate()`
- Renommer ses variable avec la fonction `rename()`
- Calcul de paramètres par sous groupes : fonctions `group_by()` et `summarise()`
- Passage du format `wide` au format `long`

#### Manipulation des variables catégorielles avec le package `forcats`

- Inspecter les variables catégorielles avec les fonctions `levels()`, `fct_count` et `fct_unique`
- Modifier l'ordre des modalités
- Modifier le nom des modalités

#### Manipuler les chaînes de caractères avec le package `stringr`

- Détection de patterns

- Découpage
- Gestion des longueurs
- Remplacement

#### Manipuler des données de date : utilisation du package `lubridate`

- Convertir les données au format YYYY-MM-DD et HH:MM:SS
- Décomposer les éléments d'année, de mois et de jour
- Calculer des différences de dates et les exprimer en jours, ou heure

#### Assemblage de tables

- Les différentes jointures (par colonne): `left join`, `right join`, `inner join` et `full join`
- Assemblage par lignes

#### Réaliser des représentations graphiques performantes avec le package `ggplot2`

- Le principe des couches successives de `ggplot2`
- Réalisation des graphiques de base : `scatterplot`, `barplots`, `line plot`, `boxplots`
- Gestion des couleurs, titres, axes et légendes
- Représentation des séries temporelles
- Utilisation du format `long` et `facetting`
- Ajouter du texte sur un graphique (ex : équation)
- Exporter son graphique : format et résolution
- Utilisation des addins `esquisse` et `Colour Picker`

#### Générer dynamiquement son rapport d'analyse avec `rmarkdown`

- Principe, formats de sorties (`html`, `docx`, `pdf`)
- Les différents éléments d'un fichier `Rmd` : `en-tête`, `chunk`,

etc...

- Gestion des éléments de texte : gras, italique, titre
- Gestion des tables
- Gestion des images
- Gestion des graphiques
- Les options des éléments de code
- Table des matières et numérotation
- Les rapports paramétrés : automatisation des rapports

d'analyse par sous groupe

- Deployer son document sur Rpubs ou bookdown

### Introduction à la programmation fonctionnelle avec le package purrr

- Les list
- Les fonctions map()
- Nested data

## Modalités

- **Type d'action** :Acquisition des connaissances
- **Moyens de la formation** :Formation présentielle – 1 poste par stagiaire – 1 vidéo projecteur – Support de cours fourni à chaque stagiaire
- **Modalités pédagogiques** :Exposés – Cas pratiques – Synthèse
- **Validation** :Exercices de validation – Attestation de stages