# Programmation lettrée

# Literate programming

04 décembre 2023

Mis à jour le 04 janvier 2024

**Table des matières**

# 1. Contexte historique

## 1.1. De la recherche reproductible…

Bien que la première apparition de l'expression "recherche reproductible" soit datée de 1992, ce concept émerge dès les années 70.

**+info -** La première apparition de l'expression "recherche reproductible" dans une publication scientifique semble être un article présenté en 1992 lors d'une réunion de la Society of Exploration Geophysics (SEG), par le groupe de Jon Claerbout à Stanford. (Claerbout & Karrenbach, 1992).

**+info -** La "recherche reproductible" est une démarche qui consiste à fournir l'ensemble des informations (texte, données, code de programmation accompagnés d'une description algorithmique) utilisées et appliquées pour obtenir des résultats présentés dans le cadre d'un travail scientifique.

À cette époque, plusieurs scientifiques, comme Jon Claerbout, s'inquiètent de l'impact des évolutions technologiques et des capacités de calcul sur la transparence et la reproductibilité des travaux scientifiques.

**+info - Jon F. Claerbout**
Géophysicien et sismologue américain né en 1938. Professeur émérite du département de Géophysique de l'Université de Stanford.
Jon Claerbout est un pionnier de l'utilisation des ordinateurs dans le traitement et le filtrage des données d'exploration sismique.
Voir sa fiche sur Wikipedia

## 1.2. … au paradigme de la programmation lettrée

Le paradigme de programmation lettrée, pensé et mis en oeuvre par Donald Knuth au cours de la même décennie (1997-78), tente d'apporter une solution à une partie du problème…

**+info - Donald Knuth**

Informaticien et mathématicien américain né en 1938.  Professeur émérite en informatique à l'université Stanford.

Il est un des pionniers de l'algorithmique et a fait de nombreuses contributions dans plusieurs branches de l'informatique théorique.

Voir sa fiche sur Wikipédia

# 2. Définition

## 2.1. En deux mots…

Le principe de la programmation lettrée est de penser l'écriture d'un programme afin qu'il soit compréhensible par l'être humain et par la machine.

**+info –** « Le paradigme de la programmation lettrée, tel qu'il a été conçu par Knuth, s'éloigne dans son écriture de l'ordonnancement imposé par l'ordinateur, et à la place autorise les développeurs à utiliser un ordre imposé par la logique et le fil de leur pensée. Les programmes lettrés sont écrits, dans un langage naturel, comme une exposition ininterrompue de la logique, à la manière d'un essai, dans lequel sont incluses des macros qui masquent les abstractions et la complexité. »
*Source : Wikipédia*

## 2.2. La programmation lettrée selon D. Knuth (1984)

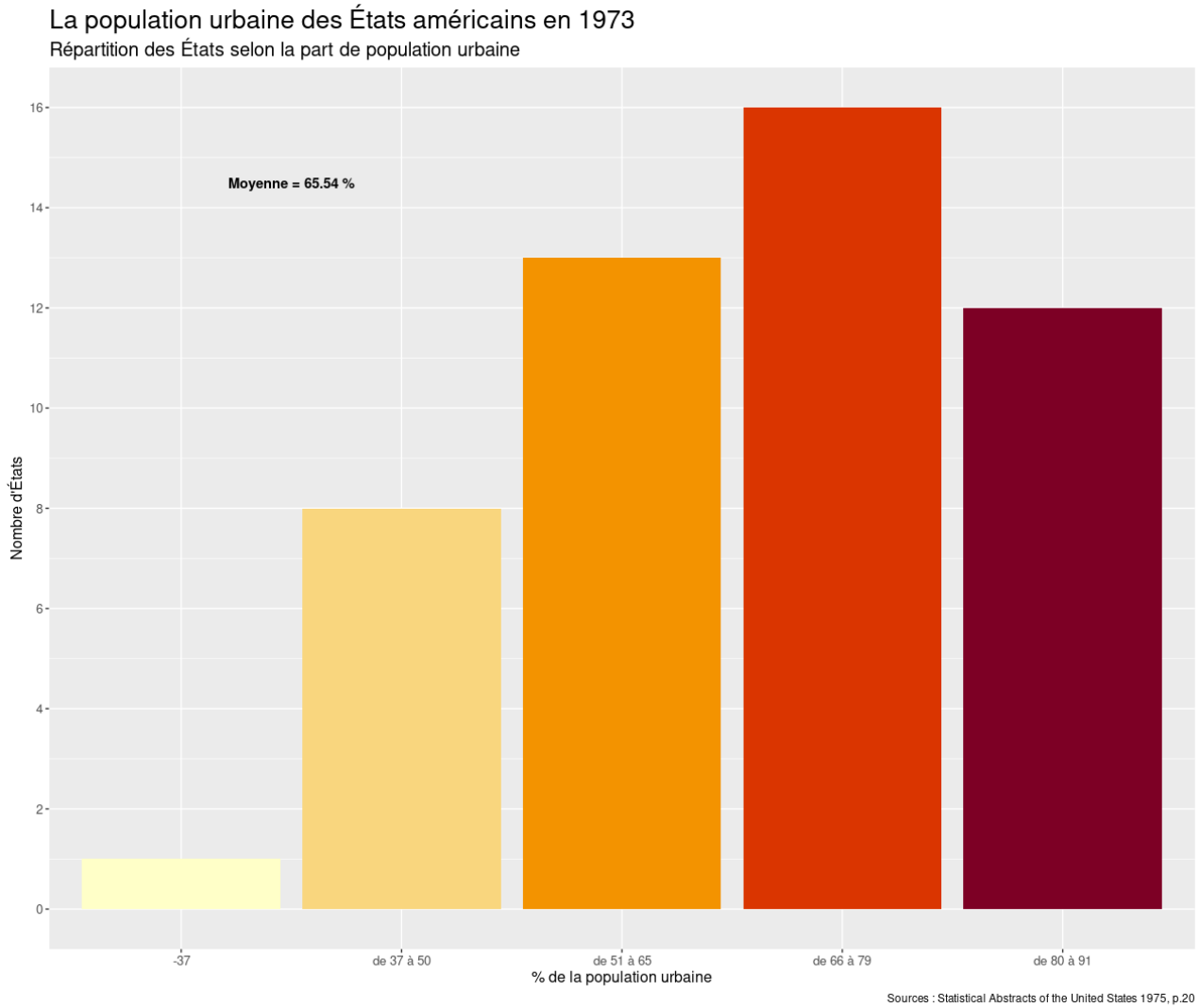
*« Nous devons changer notre attitude traditionnelle envers la construction des programmes : au lieu de considérer que notre tâche principale est de dire à un ordinateur ce qu'il doit faire, appliquons-nous plutôt à expliquer à des êtres humains ce que nous voulons que l'ordinateur fasse. (…)*
*Celui qui pratique la programmation lettrée peut être vu comme un essayiste, qui s'attache principalement à exposer son sujet dans un style visant à l'excellence. Tel un auteur, il choisit, avec soin, le dictionnaire à la main, les noms de ses variables et en explique la signification pour chacune d'elles. Il cherche donc à obtenir un programme compréhensible parce que ses concepts sont présentés dans le meilleur ordre possible. Pour cela, il utilise un mélange de méthodes formelles et informelles qui se complètent. »*
*Source : D. Knuth, « Literate Programming », The Computer Journal, British Computer Society, vol. 27, no 2, 1984, p. 97–111*
*https://academic.oup.com/comjnl/article/27/2/97/343244*

# 3. Exemple appliqué

## 3.1. De la programmation structurée…

```
library(ggplot2)
ggplot(data = USArrests,
       aes(x = cut(UrbanPop, breaks=round(c(min(USArrests$UrbanPop), mean(USArrests$UrbanPop) - 2*sd(USArrests$UrbanPop),
                                            mean(USArrests$UrbanPop) - sd(USArrests$UrbanPop), mean(USArrests$UrbanPop),
                                            mean(USArrests$UrbanPop) + sd(USArrests$UrbanPop), max(USArrests$UrbanPop)),0),
                   right  = FALSE, include.lowest = TRUE, labels = c("-37","de 37 à 50","de 51 à 65","de 66 à 79", "de 80 à 91")))) +
  labs(title = "La population urbaine des États américains en 1973", caption = "Sources : Statistical Abstracts of the United States 1975, p.20",
       subtitle = "Répartition des États selon la part de population urbaine", x = "% de la population urbaine") +
  geom_bar(fill = rev(hcl.colors(5,"YlOrRd"))) + scale_y_continuous(name = "Nombre d'États", limits = c(0, 16), breaks = seq(0,16,2)) +
  theme(plot.title = element_text(size=20), plot.subtitle = element_text(size=15), axis.text = element_text(size=10), axis.title = element_text(size=12)) +
  annotate("text", x = 1.5, y = 14.5, label = paste0("Moyenne = ", mean(USArrests$UrbanPop), " %"), fontface = "bold")
```

Les lignes de code (langage R) ci-dessus ont été pensées et optimisées pour être correctement interprétées par la machine et générer la représentation graphique suivante :



### 3.2. … à la programmation lettrée !

En appliquant le paradigme de la programmation lettrée, ces lignes de code auraient pu, tout en produisant le même résultat, s'écrire de la manière suivante :

```
#-------------------------------------------------------------------------------#
#                        Répartition des États américains,                      #
#               en fonction de leur part de population urbaine en 1973          #
#                                                                               #
#           Source : Statistical Abstracts of the United States 1975, p.20      #
#-------------------------------------------------------------------------------#

#-------------- Forme de la distribution de la variable "UrbanPop" ---------------#

# USArrests$UrbanPop = Part de la population résidant dans les zones urbaines
# Assignation de cette variable dans l'objet "ma_variable"
ma_variable <- USArrests$UrbanPop

# Statistiques descriptives
summary(ma_variable)

# Histogramme de la distribution = forme gaussienne/normale
hist(ma_variable)


#---------------- Discrétisation selon la moyenne et l'écart-type ----------------#

# Calcul des bornes de classe selon moyenne et écart-type (création d'un vecteur)
bornes_de_classe <- round(c(min(ma_variable),
                            mean(ma_variable) - 2*sd(ma_variable),
                            mean(ma_variable) - sd(ma_variable),
                            mean(ma_variable),
                            mean(ma_variable) + sd(ma_variable),
                            max(ma_variable)), 0)


# Choix des noms de classe (création d'un vecteur)
noms_de_classe <- c("-37", "de 37 à 50", "de 51 à 65", "de 66 à 79", "de 80 à 91")

# Discrétisation des individus par classe (création de la variable 'classe')
USArrests$classe  <- cut(ma_variable,
                         breaks = bornes_de_classe,
                         labels = noms_de_classe,
                         right = FALSE,
                         include.lowest = TRUE)

#------------ Représentation graphique de la distribution discrétisée ------------#

# Chargement du package 'ggplot2' pour la représentation graphique
library(ggplot2)

# Représentation graphique de la distribution
ggplot(data = USArrests, aes(x = classe)) +     # Choix de la variable à représenter
  geom_bar(fill = rev(hcl.colors(5, "YlOrRd"))) +  # Choix du type de représentation
  labs(title = "La population urbaine des États américains en 1973",
       subtitle = "Répartition des États selon la part de la population urbaine",
       caption = "Sources : Statistical Abstracts of the United States 1975, p.20",
       x = "% de la population urbaine") +
  scale_y_continuous(name = "Nombre d'États",       # Titre axe des ordonnées
                     limits = c(0, 16),             # amplitude axe des ordonnées
                     breaks = seq(0, 16, 2)) +      # labels axe des ordonnées
  annotate("text", x = 1.5, y = 14.5,               # Ajout d'une annotation
           label = paste0("Moyenne = ", mean(USArrests$UrbanPop), " %"),
           fontface = "bold")  +
  theme(plot.title = element_text(size = 20),       # Taille police titre
        plot.subtitle = element_text(size = 15),    # Taille police sous-titre
        axis.title = element_text(size = 12),       # Taille police titres axes
        axis.text = element_text(size = 10))        # Taille police labels axes
```

**FOCUS** : Utilisation de nom de variable explicite

```r
#-------------------------------------------------------------------------------#
#                          Répartition des États américains,                    #
#                en fonction de leur part de population urbaine en 1973         #
#                                                                               #
#           Source : Statistical Abstracts of the United States 1975, p.20      #
#-------------------------------------------------------------------------------#

#-------------- Forme de la distribution de la variable "UrbanPop" ---------------#

# USArrests$UrbanPop = Part de la population résidant dans les zones urbaines
# Assignation de cette variable dans l'objet "ma_variable"
ma_variable <- USArrests$UrbanPop

# Statistiques descriptives
summary(ma_variable)
```

**FOCUS** : Lignes de codes qui ne sont pas directement utiles pour la création du graphique souhaité mais qui permettent de bien retranscrire le raisonnement de l'auteur et d'expliciter ses choix.

```r
# Statistiques descriptives
summary(ma_variable)

# Histogramme de la distribution = forme gaussienne/normale
hist(ma_variable)
```

**FOCUS** : Ajout de nombreux commentaires qui aident le lecteur à comprendre rapidement le code et le raisonnement de l'auteur.

```r
# Calcul des bornes de classe selon moyenne et écart-type (création d'un vecteur)
bornes_de_classe <- round(c(min(ma_variable),
                            mean(ma_variable) - 2*sd(ma_variable),
                            mean(ma_variable) - sd(ma_variable),
                            mean(ma_variable),
                            mean(ma_variable) + sd(ma_variable),
                            max(ma_variable)), 0)
```

# 4. Programmation lettrée & *Notebooks*

## 4.1. Programmation lettrée exécutable ?

Ce paradigme n’est pas lié à un format ou à un outil.
Il est possible de mettre en pratique cette approche de la programmation dans n’importe quel type de document, même manuscrit !

Cependant, le *notebook* est sans aucun doute l’outil idéal pour mettre en œuvre ce concept, car …

Un *notebook* est une interface de programmation et d'interprétation ou exécution du code qui permet de combiner langage naturel, langage informatique et rendu computationnel (tableaux, graphiques etc.).

Le terme de *notebook* est assez populaire dans le monde anglophone, mais en raison de l’évolution et de la diversité des outils logiciels disponibles dans ce domaine, il existe de nombreuses dénominations.

**+info - Quelques appellations utilisées en français :**

- Article exécutable
- Document computationnel
- Document électronique interactif
- Cahier de programmation
- Cahier électronique d'analyse
- Calepin électronique
- Carnet de code
- Bloc-code
- Manifeste algorithmique…

### 4.2. Des pratiques du *notebook*

Plus de 50 ans après son invention, le notebook a évolué pour s'adapter à différents langages informatiques avec de nouvelles fonctionnalités. Son usage se diversifie et se généralise auprès de plus en plus de disciplines.

Comment fonctionne un *notebook* ?
Quels *notebooks* pour quels langages ?
Quels en sont les principaux usages ?
Comment optimiser sa reproductibilité et son partage...

Vous en saurez plus en consultant ce module :
https://view.genial.ly/63fde1e28e6ae700122e6a52

# 5. Références bibliographiques

Loic Desquilbet, Sabrina Granger, Boris Hejblum, Arnaud Legrand, Pascal Pernot, et al… Vers une recherche reproductible : Faire évoluer ses pratiques. Unité régionale de formation à l'information scientifique et technique de Bordeaux. Unité régionale de formation à l'information scientifique et technique de Bordeaux, pp.1-161, 2019, 979-10-97595-05-0.
https://hal.science/hal-02144142v3