# Partiels Programmation R et Mathémathiques

Jiayue Liu

31/02/2021

## Introduction

Ce rapport vise à présenter et évaluer les travaux effectués par des étudiants de la promotion MSc Data Management 2020/2022 de Paris School of Business, dans le cadre des cours *Programmation R* et *Mathématiques pour la Big Data*.

Le document est séparé en deux parties : une première concernant les travaux sur des packages en R et une deuxième portant sur les travaux en mathématiques.

L'ensemble des travaux est évalué selon les 5 critères suivants :

- La structure générale du document ;
- La présentation et la clarté rédactionnelle ;
- La qualité des codes ;
- L'apport des connaissances ;
- L'intérêt intellectuel démontré ;

Une autoévaluation des travaux auxquels nous avons contribué sera livrée à la fin du document.

## Travaux Packages R

### 1. Package `ggplot2`

- Auteur : Claude Ren
- Cliquez ici pour accéder au dossier Github décidé à ce travail

#### Synthèse

Pour contextualiser l'usage de ce package, l'auteur a d'abord comparé les fonctions et les librairies de plotting de base sous R, afin d'exposer l'apport du `ggplot2` dans la data visualization. Une série d'exemples a été donnée pour expliquer de façon concise la syntaxe. A travers un cas d'étude, nous apprenons par la suite les différentes options de personnalisation des éléments des graphiques (couleurs, police etc.) avec un focus sur des points de nuages.

**Extrait et commentaires**

```
# Initialization of the plot
g <- ggplot(cars2, aes(x=Weight, y=Displacement))
# Adding points
g <- g + geom_point(col="blue", size=3)
# Changing x and y axis
g <- g + coord_cartesian(ylim=c(100, 500), xlim=c(2500, 5000))
# Adding linear model
g <- g + geom_smooth(method="lm")
# Adding title and legends
g <- g + labs(title="Displacement by weight", subtitle="From cars2 dataset",
              y="Displacement", x="Weight", caption="cars2")
# Customize axis
g <- g + scale_x_continuous(breaks=seq(2500, 5000, 250))
# Plotting
plot(g)
```

Ces lignes de codes me semblent les plus importantes car elles permettent de générer (*plotter*) un graphique avec un minimum de personnalisation. A noter que l'opérateur `+` sert ici à ajouter des "couches" une par une sur le graphique initial. Les codes sont riches en commentaires, ce qui facilite la lecture et la compréhension des différents fonctions. Par exemple, pour les nuages de points, l'insertion d'un modèle linéaire est très facile à réaliser avec la fonction `geom_smooth()`.

**Évaluation du travail (17,4/20)**

- La structure générale du document : 17/20
- La présentation et la clarté rédactionnelle : 18/20
- La qualité des codes : 19/20
- L'apport des connaissances : 16/20
- L'intérêt intellectuel démontré : 17/20

**Conclusion**

Le travail est excellent à plusieurs titres : il contextualise bien l'usage du package en question en le comparant avec d'autres approches existantes en bien présentant ses avantages et ses inconvénients. L'explication (en anglais d'ailleurs) est très claire et fluide. C'est dommage que seuls les graphiques du type nuages de points aient été présentés en détail dans les exemples mais c'est déjà un bon début.

## 2. Package `RandomForest`

- Auteur : Thomas Massé
- Cliquez ici pour accéder au dossier Github décidé à ce travail

**Synthèse**

À l'aide d'un challenge sur Kaggle, l'auteur pu démontré l'usage du package `RandomForest` en R qui est très utilisé dans la modélisation prédictive. Le cas étudié par l'auteur consiste en un modèle de classification (vs régression) La première partie du document est consacrée à l'explication des jeux de données disponibles : il s'agit en l'occurrence des historiques des parties de jeu vidéo PUBG. La finalité du modèle construit est de prédire le taux de réussite des joueurs en fonction de ces données historiques telles que la distance parcourue,

le nombre d'armes ramassées mais aussi le résultat des parties etc. Par la suite, un travail de nettoyage a été effectué afin de garder la cohérence des données et éviter des valeurs extrêmes, une étape essentielle à la modélisation. L'auteur a également pris soins de nous montrer le modèle d'entraînement en `RandomForest`. Cette étape permet notamment d'identifier les variables explicatives estimées comme les plus importantes et d'ajuster les paramètres du modèle comme le nombre d'arbres ou le nombre de variables utilisées à chaque séparation. Enfin, le résultat de la prédiction a été exposé pour chaque type de joueurs (solo, duo, groupe).

**Extrait et commentaires**

```
#Entraînement du modèle
system.time({
set.seed(123)
solo <- randomForest(winPlacePerc ~ ., data = head(trainsetSolo, 10000),
                     na.action = na.omit, importance=T, mtry=7, ntree=200)
            })
solo
```

Cette partie de codes est importante dans la modélisation car elle fixe les paramètres de l'entraînement, lequel est essentiel à l'exactitude de résultat de sortie. La fonction `randomForest` permet de construire le modèle à partir d'un jeu de données choisi (en l'occurrence, les 10000 premières lignes de la dataframe `trainsetSolo`., avec 200 *trees* et 7 variables à chaque *split*. Dans le résultat de sortie (ici omis), on mesure l'efficacité du modèle à travers deux variables : *mean of squared residuals* et *% Var explained*.

**Évaluation du travail (16/20)**

- La structure générale du document : 15/20
- La présentation et la clarté rédactionnelle : 16/20
- La qualité des codes : 16/20
- L'apport des connaissances : 18/20
- L'intérêt intellectuel démontré : 15/20

**Conclusion**

Au travers d'un cas d'étude sur un jeu vidéo populaire, l'auteur a su démontrer l'usage du package de façon concise, structurée et concise. Cependant, les concepts adjacents du modèle ne sont pas suffisamment exposés, étant donnée que la plupart des lecteurs n'ont pas encore le niveau de connaissances sur le sujet des algorithmes d'arbres de décision. Par ailleurs, les codes ne sont pas toujours bien commentés, ce qui est dommage pour un sujet qui a l'air passionnant.

## 3. Package `sf`

- Auteur : Kabirou Kanlanfeyi & Jordy Hounsinou
- Cliquez ici pour accéder au dossier Github décidé à ce travail

**Synthèse**

Le package `sf` est une librairie en complément avec le package `sp`, tous les deux étant très utilisés pour la manipulation des données géo-spatiales sous R. Le travail présente d'abord les jeux de données utilisés - les communes d'Île de France, les arrondissements de Paris ainsi que les musées en région IdF. Par la suite, à l'aide du package `ggplot`, l'auteur a illustré de différentes fonctionnalités en `sf` : création des cartes (avec polygones, points, lignes etc.), opération des vecteurs spatiaux (notamment la mesure géométrique), vérification de la véracité des hypothèses.

**Extrait et commentaires**

```
ggplot() +
  geom_sf(data = IDF, size = 1, color = "white", fill = "black") +
  geom_sf(data = musees, size = 1, color = "red", fill = "black") +
  ggtitle("Localisation des musées en île de France") +
  theme(plot.title = element_text(hjust = 0.5)) +
  coord_sf()
```

Les fonctions du package `sf` peuvent être directement imputées au package `ggplot` afin de visualiser les données géographiques. L'usage de l'opérateur `+` y est particulièrement utile. A la même manière que 'leaflet', cela suit une logique d'ajout des couches qui contiennent des jeux de données différents. Il existe également de différentes options de personnalisation graphiques, mais qui sont faciles à maîtriser d'un point de vue syntaxique.

**Évaluation du travail (17/20)**

- La structure générale du document : 18/20
- La présentation et la clarté rédactionnelle : 18/20
- La qualité des codes : 17/20
- L'apport des connaissances : 17/20
- L'intérêt intellectuel démontré : 16/20

**Conclusion**

Ce travail est très complet dans la mesure où il expose une grande partie des fonctionnalités prises en charge par le package en question. Les explications fournies font preuve de clarté et de précision. Les codes ne sont pas garnis commentaires mais cela n'aurait pas d'impact sur la qualité globale car les lignes de commandes sont souvent intuitives et faciles à comprendre. Un excellent travail dans l'ensemble!

## 4. Package `plotly`

- Auteur : Imen Derrouiche & Olfa Lamti
- Cliquez ici pour accéder au dossier Github décidé à ce travail

**Synthèse**

Le package `plotly` est très pratique pour la data visualisation interactive, notamment grâce à ses multifonctionnalités en termes de création graphiques mais aussi à la barre d'interaction dans les graphiques générés. Ainsi, l'auteur a su montrer de différents types de visuels que l'on peut créer avec ce package (nuages de points, lignes et polygones, diagrammes, surfaces 3D etc.) ainsi que les différents manières d'organisation visuelle (c'est-à-dire des sous-graphiques). Enfin, ces contenus visuels peuvent être exportés dans un fichier HTML et être publiés.

**Extrait et commentaires**

```
# On affecte à "plotlist" une fonction où on entre le nombre de graphique
plotList <- function(nplots) {
  lapply(seq_len(nplots),
         function(x) plot_ly())
```

```
}

#On appelle la fonction et on crée 6 graphiques à deux lignes, qui partagent les mêmes axes.
s1 <- subplot(plotList(6),
              nrows = 2,
              shareX = TRUE,
              shareY = TRUE)

#On appelle la fonction et on crée 2 graphiques à deux lignes, qui partagent le même axe d'ordonnées.
s2 <- subplot(plotList(2),
              shareY = TRUE)

#On affiche le graphique sur 3 lignes, avec un espacement de 0,04
subplot(
  s1,
  s2,
  plot_ly(),
  nrows = 3,
  margin = 0.04,
  heights = c(0.6, 0.3, 0.1)
)
```

Le package `plotly` permet non seulement de combiner différents graphiques dans un seul espace, mais surtout de créer des graphiques récursifs, c'est à dire d'affecter les mêmes axes à un ensemble de sous-graphiques. Suivant les commentaire de l'auteur, on crée d'abord une fonction qui permet de saisir le nombre de graphiques affichés. Par la suite, on construit des sous-graphiques avec la fonction `subplot` (ici `s1` et `s2`) pour ensuite les ordonner dans un seul espace.

**Évaluation du travail (17,4/20)**

- La structure générale du document : 19/20
- La présentation et la clarté rédactionnelle : 18/20
- La qualité des codes : 18/20
- L'apport des connaissances : 16/20
- L'intérêt intellectuel démontré : 16/20

**Conclusion**

Grâce au format HTML du contenu produit, ce tutoriel a présenté les différents aspects du package de manière bien organisée et très claire. L'auteur a également étudié en détail comment créer de divers graphiques en utilisant les fonctions du `plotly`. Par ailleurs, les codes sont enrichis de commentaires concis mais très pertinents, ce qui rend la lecture assez agréable. Un travail de grande qualité.

## 5. Package `shiny`

- Auteur : Rindra Lutz & William Robache
- Cliquez ici pour accéder au dossier Github décidé à ce travail

**Synthèse**

Pour découvrir l'usage du package `shiny`, l'auteur de ce tutoriel a proposé un exemple concret qui consiste à créer une interface Web, dans laquelle l'utilisateur peut saisir le nombre de lignes qu'il souhaite à partir d'un

jeu de données. En effet, en `shiny`, on suit toujours une logique d'interface (la partie visible et visualisée de l'application) et de serveur (le traitement des données saisies et sorties). Sans donner d'exemple concret, l'auteur a proposé également d'explorer davantage de possibilités avec `shiny` pour ceux qui maîtrisent les langages de web design et les balises.

**Extrait et commentaires**

```
server <- shinyServer(function(input, output) {
  #On retrouve ici l'élement "dataset", qui communique avec ui par 'output'.
  output$dataset <- renderTable({
    #Nous imprimons les éléments d'après les données en entrée : ces derniers sont appellés avec `input
    if(input$labs == "cars"){
      print(head(cars, input$lignes))
    } else if(input$labs == "rock"){
      print(head(rock, input$lignes))
    } else if(input$labs == "sleep"){
      print(head(sleep, input$lignes))
    } else {
      print(head(beaver1, input$lignes))
    }
  })
})
```

Cette partie de codes permet de construire le serveur de notre application. En fonction des données saisies (*input*) dans l'interface par l'utilisateur (2 variables : `labs` et `lignes`), le serveur fait sortir des données correspondant aux commandes pré-définies (*output*). Il est à noter que toutes ces commandes devraient être incluses à l'intérieur de la fonction `renderTable`.

**Évaluation du travail (15,6)**

- La structure générale du document : 16/20
- La présentation et la clarté rédactionnelle : 17/20
- La qualité des codes : 17/20
- L'apport des connaissances : 15/20
- L'intérêt intellectuel démontré : 13/20

**Conclusion**

Le travail réalisé permet de bien comprendre la logique et le fonctionnement derrière une application Web `shiny` en interaction avec l'utilisateur en temps réel. Les codes sont commentés dans les détails, ce qui facilite la compréhension pour les débutants. Cependant, malgré la qualité didactique, on peut tout de même regretter le manque d'applications plus complexes qui auraient mobilisé des connaissances plus transversales.

# Travaux mathématiques

## 1. Modèles de régression

- Auteur : Nina Zoumanigui
- Cliquez ici pour accéder au dossier Github décidé à ce travail

#### Synthèse

La régression statistique permet de modéliser les phénomènes naturels ou sociaux à partir d'une série de variables d'intérêt : d'un côté les variables dépendantes qui quantifient ou qualifient la conséquence, et de l'autre les variables indépendantes qui mesurent ou décrivent les différentes causes (variables explicatives). On peut faire une régression linéaire (simple ou multiple) en partant d'une hypothèse nulle qui consiste à dire qu'il n'y a pas de relation significative entre la variable dépendante et la ou les variable(s) indépendantes.

#### Extrait et commentaires

```
$$yi = b0 + b1xi1 + b2xi2 + · · · + bpxip + ei , i = 1, · · · , n$$
```

$$yi = b0 + b1xi1 + b2xi2 + ůůů + bpxip + ei, i = 1, ůůů, n$$

Cette formule a pour but de montrer la construction d'une régression linéaire multiple. Pour $i = 1, \ldots, n$, nous avons $y_i$ est censé être la variable dépendante tandis que les variables explicatives sont dénotées $x_i$. $b_0$ est censé être l'intersection et $\epsilon_i$ représente les résidus statistiques. Néanmoins, cette partie de code n'a pas suivi la syntaxe de LaTeXet n'est pas non plus commentée.

#### Évaluation du travail (11,4/20)

- La structure générale du document : 12/20
- La présentation et la clarté rédactionnelle : 12/20
- La qualité des codes : 10/20
- L'apport des connaissances : 12/20
- L'intérêt intellectuel démontré : 11/20

#### Conclusion

Ce document est censé nous expliquer les notions mathématiques derrière les modèles de régression, notamment la régression linéaire. Cependant, nous constatons que l'auteur n'a pas pu rentrer dans les détails et que les équations mathématiques sont malheureusement dépourvues de clarté et de rigueur.

## 2. Régression logistique

- Auteur : Gaspard Palay
- Cliquez ici pour accéder au dossier Github décidé à ce travail

**Synthèse**

Ce document cherche à explorer les fondements mathématiques de la régression logistique et à illustrer son application à travers un cas d'études sur la prédiction de la survie des voyageurs sur le Titanic. À la différence de la régression linéaire, le régression logique sert à étudier la probabilité d'une variable binaire (VRAI/FAUX) à partir des variables explicatives. Cela suppose que nous avons besoin de construire un modèle de régression mais en parallèle il nous faut trouver la distribution bayésienne des deux conséquences.

**Extrait et commentaires**

```
$$ln\bigg[\frac{P(X/Y = +)}{P(X/Y = -) } \bigg] = b_{0}+b_{1}+ ... +b_{j}X_{j}$$
```

$$ln\left[\frac{P(X/Y = +)}{P(X/Y = -)}\right] = b_0 + b_1 + ... + b_j X_j$$

Cette formule est fondamentale pour comprendre la régression logique. Ici, la probabilité que le résultat soit vrai est dénotée par $P(X \mid Y = +)$ tandis que celle que le résultat soit faux est dénotée $P(X \mid Y = -)$. La fonction logit ($ln$) permet de calculer les *odds*, soit le ratio de ces deux probabilités. Les variables situées à la partie droite de l'équation constituent les variables explicatives.

**Évaluation du travail (14,4/20)**

- La structure générale du document : 15/20
- La présentation et la clarté rédactionnelle : 14/20
- La qualité des codes : 13/20
- L'apport des connaissances : 16/20
- L'intérêt intellectuel démontré : 14/20

**Conclusion**

Comme évoqué dans le document, la qualité du travail effectué est certes limitée par les connaissances mathématiques de l'auteur. Le but de cet exercice est néanmoins atteint car nous avons pu, à travers la lecture, se donner une idée de ce que c'est une régression logique. Le cas d'études est également très intéressant et permet de voir la finalité réelle de ces notions parfois très abstraites.

## 3. K-Nearest Neighbors (KNN)

- Auteur : Antoine Serreau, Benjamin Guigon & Corentin Bretonnière
- Cliquez ici pour accéder au dossier Github décidé à ce travail

**Synthèse**

La méthode KNN (K-Nearest Neighbors) consiste à estimer la valeur d'un point de données inconnu à partir d'autres points existants qui lui ressemblent le plus - d'où le concept des voisins les plus proches. Pour ce faire, il convient de calculer les distance entre le point d'intérêt et tous les points voisins. Plusieurs méthodes de calcul sont possibles, mais la distance euclidienne semble être un bon choix pour les données quantitatives du même type. Ensuite on en choisit $k$ points les plus proches (ayant donc le plus de similarité) pour calculer la moyenne de ces valeurs. Afin d'obtenir la valeur de $k$ optimale, il est conseillé de faire tourner plusieurs fois le modèle KNN et comparer le taux d'erreurs.

**Extrait et commentaires**

```
$$D_e(x,y) = \sqrt{\sum_{j=1}^{n}(x_j-y_j)^2}$$
```

$$D_e(x,y) = \sqrt{\sum_{j=1}^{n}(x_j-y_j)^2}$$

La première équation sert à calculer la distance euclidienne entre deux points. Il s'agit tout simplement de calculer la racine carrée de la somme des différences carrées entre les coordonnées ($x_j$ et $y_j$) entre deux points. Grâce à cette valeur, on pourra par la suite identifier les points les plus proches de notre point d'intérêt et ensuite l'affecter à une classe / un label.

**Évaluation du travail (15,4/20)**

- La structure générale du document : 16/20
- La présentation et la clarté rédactionnelle : 16/20
- La qualité des codes : 15/20
- L'apport des connaissances : 16/20
- L'intérêt intellectuel démontré : 14/20

**Conclusion**

Cet article bien documenté permet d'avoir une vision globale sur la méthode KNN. L'ensemble des concepts sont bien expliqués, avec des exemples concrets à simples. L'auteur a également pris soins de souligner les avantages et inconvénients de cette méthode pour avertir les lecteurs sur ses cas d'usage et ses limites.

## 4. Marche aléatoire

- Auteur : William Robache & Marko Arsic
- Cliquez ici pour accéder au dossier Github décidé à ce travail

**Synthèse**

Au travers d'un exemple de l'algorithme PageRank de Google, l'auteur nous a expliqué le concept et l'usage de la marche aléatoire. Celle-ci consiste à quantifier les pas aléatoires effectués par un agent dans un espace à dimension variée. L'ensemble des pas forment une chaîne dite ergodique qui est caractérisée par son irréductibilité, sa récurrence positive et sa non-périodicité. Google utilise une combinaison des chaînes de Markov et de la marche aléatoire pour déterminer la notorité d'une page web.

**Extrait et commentaires**

```
Soit $d\geq 1$ et soit $(e_1,\dots, e_d)$ la base canonique de $\mathbb{Z^d}$.
Soit $(X_i)$ une suite de variable aléatoires indépendantes à valeurs dans
$\{(\pm e_1,\dots,\pm e_d)\}$. On appelle marche aléatoire associée la suite
de variables aléatoires $(S_n)_{n\geq 1}$ où $S_n$ est défini par $$
S_n=X_1+\dots+X_n. $$


Lors de l'étude des marches aléatoires, nous souhaitons savoir si le système
```

```
revient à son point de départ, c'est-à-dire s'il revient à son origine.

On dit que la marche alétoire est récurrente si:

$$ P(S_n=0\textrm{ infiniment souvent})=1.$$
On dit que la marche aléatoire est transiente si:

$$ P(S_n=0\textrm{ infiniment souvent})=0. $$
On considère une marche aléatoire sur $\mathbb Z^d$ telle que $$P(X_i=e_j)=P(X_i=-e_j)=\frac 1{2d }$$
pour tout $i\geq 1$ et tout $j\in\{1,\dots,d\}$.
Alors:
- La marche aléatoire est récurrente si $d=1$ ou $d=2$
- La marche aléatoire est transiente si $d\geq 3$
```

Cette partie du texte explique les concepts fondamentaux de la marche aléatoire. La suite $S_n$ représente la somme des pas aléatoires effectués. Puisque les variables sont tirées aléatoirement et sont à deux états possibles, on observe une convergence de la suite $S_n$ vers l'état neutre (ici c'est-à-dire la valeur 0). Si cela est le cas, c'est donc une marche aléatoire récurrente. Au contraire, si après un certain nombre de pas, la suite ne tend pas vers 0, alors on peut l'appeler une marche aléatoire transciente.

#### Évaluation du travail (13,6/20)

- La structure générale du document : 11/20
- La présentation et la clarté rédactionnelle : 11/20
- La qualité des codes : 14/20
- L'apport des connaissances : 16/20
- L'intérêt intellectuel démontré : 16/20

#### Conclusion

Le travail réalisé donne quelques éléments clé pour comprendre la logique d'une marche aléatoire ainsi que son implication en algorithmique. Le niveau de mathématiques étant limité, je n'ai pas pu approfondir mon appréciation de certaines notions comme l'ergodicité. Ceci dit, l'impression générale est que la qualité rédactionnelle est en-dessous de la moyenne, avec certaines expressions mathématiques mal affichées.

## 5. K-means ++

- Auteur : Lucas Billaud
- Cliquez ici pour accéder au dossier Github décidé à ce travail

#### Synthèse

La méthode K-means est une technique de machine learning non-supervisé qui consiste à créer des clusters parmi les points de données. Pour cela, un centroïde est désigné pour un ensemble de points jugés similaires. Ensuite on minimise la distance entre les points et le centroïde afin de déterminer la classification (*clustering*). La distance euclidienne reste l'une des méthodes de calcul les plus courantes, mais d'autres méthodes, notamment la distance Levenshtein peut s'avérer utile pour d'autres cas d'usage (notamment en linguistique). L'auteur a également tenté d'explorer un papier de recherche qui propose une version améliorée de l'algorithme : K-means ++. Celle-ci utilise une ature manière pour placer les centroïdes supposés.

**Extrait et commentaires**

$$\Theta = \sum_{x \in X} \min \|x - c\|^2$$

$\Theta$ est donc une fonction qui minimise la distance euclidienne entre les points $(x)$ dans l'espace donné $(X)$ et le centroïde $(c)$. Si les distances mesurées peuvent encore dimunuer, alors les clusters ne sont pas encore stabilisés : on devrait donc essayer d'autres centroïdes. (Cette formule est extraite du document ici étudié mais ne provient pas d'une syntaxe LaTeX).

**Évaluation du travail (12,8/20)**

- La structure générale du document : 12/20
- La présentation et la clarté rédactionnelle : 12/20
- La qualité des codes : 13/20
- L'apport des connaissances : 14/20
- L'intérêt intellectuel démontré : 13/20

**Conclusion**

Ce document apporte une explication détaillée sur la méthode K-means. En raison de compétences mathématiques plus avancées, l'auteur n'a pas pu démontrer les principes mathématiques de K-means ++. Par ailleurs, les équations citées ne sont pas très bien structurées et présentent quelques défauts d'affichage.

# Auto-évaluation des travaux R et mathématiques

## 1. Package `dplyr`

- Auteur : Soukaina El Ghaldy & Jiayue Liu
- Cliquez ici pour accéder au dossier Github décidé à ce travail

Ce travail permet aux lecteurs d'avoir une vision globale non seulement sur le package lui-même mais aussi sur l'univers `tidyverse`, ce qui est utile pour les débutants. Les exemples donnés illustrent de façon claire les fonctions permises dans le package. En revanche, lorsqu'on vérifie de nouveau le document, on peut s'apercevoir quelques petites erreurs dans les codes.

**Évaluation du travail (16/20)**

- La structure générale du document : 17/20
- La présentation et la clarté rédactionnelle : 16/20
- La qualité des codes : 16/20
- L'apport des connaissances : 15/20
- L'intérêt intellectuel démontré : 16/20

## 2. Modèles spatio-temporels

- Auteur : Jiayue Liu
- Cliquez ici pour accéder au dossier Github décidé à ce travail

Le document a pour but d'exposer les implications mathématiques derrière le package `inla` qui est l'une des méthodes courantes dans la modélisation spatio-temporels. Au travers d'un exemple concret, nous voulions également démontrer l'usage de ce package dans le contexte de la santé publique. Cependant, faute de compétences approfondies sur la probabilité bayésienne et la méthode Laplace, il nous a été difficile de saisir toutes les nuances et subtilités de cette approche.

**Évaluation du travail (14,8/20)**

- La structure générale du document : 15/20
- La présentation et la clarté rédactionnelle : 13/20
- La qualité des codes : 16/20
- L'apport des connaissances : 15/20
- L'intérêt intellectuel démontré : 15/20