

Comment rédiger un rapport scientifique
reproductible avec **knitr**, **RMarkdown** et plein
d'autres amis

Pr JEAN R. LOBRY

Cette fiche donne quelques indications pour rédiger un rapport reproductible intégrant automatiquement des analyses statistiques et des graphiques produits par . On essaye de baliser ici une randonnée d'initiation à travers les différents langages de balisage utiles à l'accomplissement de cet objectif. Il s'agit ici plus des grandes lignes d'un programme d'exploration que d'un document visant à l'exhaustivité.

Contents

1	Introduction	3
1.1	Écrire un rapport	3
1.2	Approche artisanale : copier/coller manuel	3
1.3	La programmation littéraire	4
1.4	Les systèmes de gestion de version	7
2	Aide SVP Urgent !!!	7
2.1	Les forums de discussion	7
2.2	Qu'est-ce qu'un code reproductible ?	8
2.3	Comment insérer des données dans un message	10
2.4	Comment détailler son environnement de travail	11
2.5	L'étiquette [Résolu]	12
3	Approche professionnelle : utilisation de knitr et RMarkdonw	12
3.1	Qu'est ce que knitr ?	12
3.2	Qu'est ce que RMarkdown ?	12
3.3	Qu'est-ce que pandoc ?	18
3.4	Pré-requis : installer un paquet 	19
3.4.1	Paquets ne nécessitant pas d'être compilés	19
3.4.2	Paquets nécessitant d'être compilés	20
3.4.3	Paquets nécessitant un programme externe	21

3.5	Pré-requis : installer \LaTeX	21
3.5.1	La distribution $\text{Tiny}\LaTeX$	21
3.5.2	Un petit test	22
3.5.3	Mise à jour	23
3.5.4	Pré-visualisation	23
3.5.5	Le mode visuel	23
3.6	Un exemple $\text{R}\text{Markdown}$ minimaliste	24
3.7	Dossier de travail et reproductibilité	26
3.8	Séparation de la forme et du fond	27
3.9	Choix du format de sortie	29
3.10	Structuration du document	31
3.11	Gérer les références bibliographiques	32
3.11.1	Importance des références bibliographiques	32
3.11.2	Principe de surjectivité	33
3.11.3	Le format $\text{Bib}\LaTeX$	33
3.11.4	Première et seconde main	35
3.11.5	Nomenclatures particulières	36
4	Tricotage	38
4.1	Insérer du code \mathbb{R}	38
4.2	Récupérer le résultat d'un calcul \mathbb{R}	45
4.3	Insérer un tableau	46
4.4	Insérer une figure	46
5	Les langages de balisage	48
5.1	Pas YAML	48
5.2	\LaTeX	48
5.3	HTML	48
5.4	$\text{R}\text{Markdown}$	48
	Références	48

1 Introduction

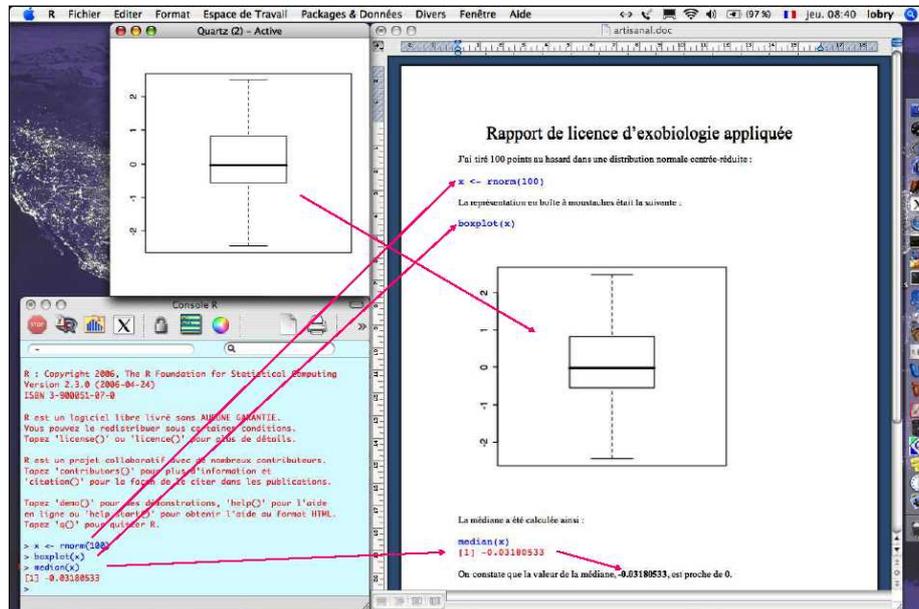
1.1 Écrire un rapport

IL existe deux grandes approches pour rédiger un rapport faisant appel à des analyses et à des graphiques produits sous **R**.

1. Une première approche, que nous qualifierons d'artisanale, sans aucun sens péjoratif, est destinée à la production d'un nombre très restreint de rapports tous très courts. Cette approche est adaptée si vous n'envisagez de produire que des rapports de façon sporadique et que vous n'êtes pas intéressé par une carrière scientifique au sens large du terme.
2. Une deuxième approche, que nous qualifierons de professionnelle, est plus adaptée à la production de nombreux rapports, ou d'un rapport de taille conséquente. Elle consiste à intégrer directement le code **R** dans le rapport et à utiliser **R** pour produire les graphiques et les analyses à intégrer directement dans le document final (oui c'est possible !). Ceci garanti la parfaite *reproductibilité* de vos résultats.

1.2 Approche artisanale : copier/coller manuel

CETTE approche consiste simplement à ouvrir simultanément **R** et votre traitement de texte favori pour copier/coller les résultats qui vous intéressent :



Voici quelques indications utiles pour obtenir un résultat acceptable :

- ★ Pour les graphiques, commencez par re-dimensionner la fenêtre graphique dans **R** jusqu'à ce que vous soyez satisfait du résultat. Copier/coller

ensuite le graphique dans votre traitement de texte, puis re-dimensionnez le, de façon proportionnelle en x et y pour ne pas introduire de distorsions, pour obtenir la taille finale désirée.

- ★ Pour les commandes `R` entrées dans la console, copiez/collez les dans le traitement de texte puis supprimez le caractère d'invite de commande (par défaut le caractère `'>'`) de la console `R`. Ceci vous permettra ultérieurement de facilement copier/coller dans l'autre sens, de votre traitement de texte vers `R`. Définissez un style pour mettre en évidence qu'il s'agit de commandes `R`, par exemple en utilisant une police de caractères **non proportionnelle**, éventuellement de **couleur rouge** pour signifier qu'il s'agit de commandes `R` en entrée.
- ★ Pour les résultats obtenus dans la console, définissez un style similaire à celui des entrées, mais changez la couleur, par exemple en **couleur bleue**, pour mettre en évidence qu'il s'agit de résultats, et non d'entrées.
- ★ Pour intégrer des résultats numériques dans le corps du texte, il vous faudra faire un copier/coller de plus, veillez à maintenir la cohérence de l'ensemble !

CETTE approche permet de générer facilement de petits documents de qualité, mais elle trouve très rapidement ses limites car il est extrêmement fastidieux de maintenir la cohérence entre les différents éléments du rapport. Supposez par exemple que vous ayez envie de modifier un graphique, pour par exemple lui rajouter un titre. Il vous faudra alors générer le graphique modifié sous `R`, mais comme vous avez oublié de noter quelque part la dimension du périphérique utilisé dans la version précédente, vous allez perdre du temps pour retrouver un résultat acceptable. Puis il faudra copier/coller le graphique dans le traitement de texte, et de nouveau, le re-dimensionner à la main (et encore une fois vous n'avez pas noté ce que vous aviez fait la dernière fois, encore du temps perdu à tâtonner pour retrouver quelque chose d'à peu près similaire). Mais ce n'est pas fini : le code `R` n'est plus cohérent avec le graphique, il faut une nouvelle opération manuelle de copier/coller, et de remise en forme dans le traitement de texte, pour assurer la cohérence. Et, bien entendu, si vous avez intégré des résultats numériques dans le corps du texte, il faudra une troisième opération manuelle de copier/coller pour mettre tout le document en cohérence. On arrive très rapidement aux limites de l'exercice.

1.3 La programmation littéraire

TOUT a commencé en 1984 quand Donald Knuth a introduit le concept de *programmation littéraire* [50]. Dans un exercice de programmation classique vous expliquez à l'ordinateur ce que vous voulez qu'il fasse pour vous. Vous pouvez commenter votre code, par exemple ci-dessous tout ce qui suit le caractère `#` sera ignoré par l'ordinateur, pour aider un lecteur humain à comprendre ce que vous avez essayé de faire :

```
factorielle <- fonction(n){
  n <- as.integer(n)          # users are losers
  if(n < 0) stop("n < 0")    # idem
  if(n == 0) return(1)        # by def
  return(n*factorielle(n - 1)) # by rec
}
```

DONALD Knuth propose d'inverser les choses et de considérer qu'un programme est une œuvre littéraire dans laquelle vous essayez d'expliquer à autre être *humain* comment vous avez tenté de vous faire obéir par un ordinateur. Cela pourrait ressembler à ceci :

```
Le sujet du devoir à la maison était de proposer une fonction robuste
pour calculer la factorielle d'un nombre entier n. La première chose
à laquelle j'ai pensé c'est de forcer l'argument de ma fonction à
être un entier :-
-
<<pre1,eval=FALSE>>=-
· n <- as.integer(n)
@-
-
La seconde chose est que la factorielle n'est définie que pour des
entiers positifs ou nuls. J'ai donc décidé d'arrêter ma fonction avec
un message d'erreur si jamais ce n'était pas le cas :-
-
<<pre2,eval=FALSE>>=-
· if(n < 0) stop("un entier positif ou nul est attendu pour n")
@-
-
J'ai traité ensuite le cas particulier de la factorielle de zéro qui vaut 1 :-
-
<<post1,eval=FALSE>>=-
· if(n == 0) return(1)
@-
-
J'ai enfin utilisé la relation de récurrence pour calculer la factorielle de n :
-
<<post2,eval=FALSE>>=-
· return(n*factorielle(n-1))
@-
-
Au final ma fonction est définie par :-
-
<<factodef>>=-
factorielle <- function(n){
<<pre1>>
<<pre2>>
<<post1>>
<<post2>>
}
@-
-
J'ai lu dans wikipedia que la factorielle de 6 vallait 720, j'ai vérifié sur
ce cas particulier que ma fonction donnait le bon résultat :-
-
<<>>=-
factorielle(6)
@-
```

COMME on peut le constater ici, le code du programme est passé en quelque sorte en *commentaire* de votre document. Mais ce sont des commentaires particuliers dans le sens où ils sont compréhensibles par l'ordinateur. On peut donc lui demander de les exécuter pour produire le document final :

Le sujet du devoir à la maison était de proposer une fonction robuste pour calculer la factorielle d'un nombre entier n . La première chose à laquelle j'ai pensé c'est de forcer l'argument de ma fonction à être un entier :

```
n <- as.integer(n)
```

La seconde chose est que la factorielle n'est définie que pour des entiers positifs ou nuls. J'ai donc décidé d'arrêter ma fonction avec un message d'erreur si jamais ce n'était pas le cas :

```
if(n < 0) stop("un entier positif ou nul est attendu pour n")
```

J'ai traité ensuite le cas particulier de la factorielle de zéro qui vaut 1 :

```
if(n == 0) return(1)
```

J'ai enfin utilisé la relation de récurrence pour calculer la factorielle de n :

```
return(n*factorielle(n - 1))
```

Au final ma fonction est définie par :

```
factorielle <- function(n){  
  n <- as.integer(n)  
  if(n < 0) stop("un entier positif ou nul est attendu pour n")  
  if(n == 0) return(1)  
  return(n*factorielle(n - 1))  
}
```

J'ai lu dans wikipedia que la factorielle de 6 valait 720, j'ai vérifié sur ce cas particulier que ma fonction donnait le bon résultat :

```
factorielle(6)
```

```
[1] 720
```

NOUS n'insisterons pas plus sur la syntaxe `noweb` [81] illustrée ici parce que ce n'est pas celle que vous utiliserez pour rédiger votre rapport. On notera simplement avec amusement qu'on la voit parfois pointer le bout de son nez, par exemple quand on veut ré-utiliser un fragment de code (*cf* paragraphe 4.1 page 38).

LES outils développés pour faire de la programmation littéraire ont rapidement été adoptés par les scientifiques soucieux d'assurer la reproductibilité de leurs résultats. Pour que vos lecteurs puissent facilement reproduire vos résultats il faut leur donner accès non seulement aux données mais aussi aux méthodes que vous avez utilisées. C'est donc un exercice de programmation littéraire sur un jeu de données. Ne croyez pas que ce soit chose aisée, Buckheit et Donoho, dans un article [15] célèbre et plein d'humour, ont donné des exemples de leurs propres déboires :

1. *Burning the Midnight Oil*. Pétage de plombs après une journée de travail de 18 heures pour retrouver la bonne version imprimée des figures pour un article en comptant 30.
2. *The Stolen Briefcase*. Impossible de reproduire les figures originales per-

dues suite au vol d'une valise, abandon du manuscrit.

3. *Who's on First?* Difficultés de communication entre un professeur et son étudiant faute d'outils pour faciliter la reproductibilité des résultats.
4. *A Year is a Long Time in this Business.* Impossible d'appliquer une méthode à un nouveau jeu de donnée seulement un an après sa publication.
5. *A la Recherche des Parametres Perdues [sic].* Impossible de retrouver la bonne combinaison de valeurs de paramètres pour reproduire la figure d'un article.

1.4 Les systèmes de gestion de version

NOUS ne parlerons pas ici des outils facilitant la gestion des différentes versions d'un document, ou d'un ensemble de documents. Ces outils sont indispensables pour tout travail collaboratif. Les plus connus sont `subversion` et `GitHub`. Ils sont dispensables pour la rédaction d'un rapport dont vous êtes l'unique auteur, encore que, même dans ce cas de figure, certains préfèrent s'appuyer sur de tels systèmes pour faciliter la gestion des différentes versions de leurs documents.

2 Aide SVP Urgent !!!

LES ordinateurs étant particulièrement psychorigides, on gagne rarement contre eux à jouer au plus têt. C'était déjà illustré avec humour du temps des premiers langages de programmation comme le FORTRAN :

```
fortunes::fortune("FORTRAN")
This reminds me of a famous FORTRAN code snippet:
10 STOP
STOP
STOP
! IN CASE STILL SKIDDING
GOTO 10
-- Carl Witthoft (in response to the question how to completely stop R scripts
after stop())
R-help (January 2011)
```

2.1 Les forums de discussion

QUAND on rencontre un problème et que tout a échoué (consultation de la documentation, recherches frénétiques avec son moteur de recherche favori) rien ne vaut que de demander assistance à un autre être *humain*. Les forums de discussion spécialisés sont à coup sûr la meilleure façon de solliciter de l'aide. On évitera bien entendu de mettre en sujet de sa question le titre de cette section mais quelque chose de plus *informatif* pour le lecteur.

POUR illustrer tout l'intérêt des forums de discussion on peut citer Yihui Xie qui à propos de la distribution TinyTeX [104] déclare¹ : « [w]hy do I exclude the documentations? Tell me honestly: how many times have you found a solution via StackExchange², and how many times have you tried to

¹<https://yihui.org/tinytex/faq/>

²Un forum de discussion anglophone qui dispose d'un salon dédiée à TeX et LaTeX : <https://tex.stackexchange.com/>

read the package documentation? Even with the full documentation installed, you probably don't even know where to find these documentation files on your computer. The documentation files take a lot of disk space, and I believe they are rarely read by an average user, so they are not included. The address bar of your web browser is the most convenient documentation: type and search. »

L'OBJECTIF de ce document est de baliser une randonnée d'initiation à travers les différents langages de balisage utiles à la production d'un rapport reproductible. Nous espérons ainsi qu'en cas de problème vous saurez à quelle communauté vous adresser pour demander de l'aide. Est-ce que j'ai un problème de \mathbb{R} , de \LaTeX ou plutôt d'HTML ? Si vous savez déjà vers qui vous tourner, vous n'aurez pas perdu votre temps ici. Mais comme l'objectif est de produire un document reproductible, nous espérons que vous apprendrez au passage, pour le même prix, à poser de façon *efficace* vos questions dans les forums de discussion.

POUR illustrer ce propos, nous reprenons ci-après les conseils donnés par François Bonnot dans le forum *Groupe des utilisateurs du logiciel R*³. C'est un forum francophone d'échange autour du logiciel de calcul statistique \mathbb{R} , mais pas un forum sur l'apprentissage des méthodes statistiques. Si vous avez une question sur \mathbb{R} vous pouvez y poster, si vous avez une question de statistique (genre : c'est quoi la p-value ?) vous allez vous faire éjecter, ou pire ignorer. C'est vrai, d'une façon générale, pour tout forum de discussion : on ne pose pas une question à propos de \LaTeX dans un forum de BDSM, sauf à être un masochiste particulièrement retors.

2.2 Qu'est-ce qu'un code reproductible ?

CETTE section est directement adaptée des recommandations de François Bonnot du 2015-10-19⁴ sur le forum francophone des utilisateurs du logiciel \mathbb{R} .

LES questions posées sur le forum concernent fréquemment une partie de code insérée dans le corps du message. Malheureusement il est souvent difficile de répondre à la question parce que le code inséré n'est pas reproductible. Cela a pour conséquence non seulement d'alourdir la discussion par des messages hors du sujet technique (« Quel paquet utilisez-vous ? », « Je n'arrive pas à reproduire votre sortie », etc.), mais aussi de décourager les membres du forum qui sauraient répondre mais n'ont pas de temps à perdre à deviner ce qui manque dans le code. Pour s'en persuader, il suffit de rechercher le terme « reproductible » sur le forum.

QU'EST-CE qu'un code reproductible ? Un code reproductible est un code dont l'exécution produit sur tous les ordinateurs une sortie identique à celle produite sur l'ordinateur de celui qui pose la question, sans nécessiter l'exécution préalable d'autres lignes de code. Dans la plupart des cas, le code est non reproductible parce qu'il y manque des données ou l'indication d'un paquet.

Exemple 1 *non* reproductible:

```
S <- matrix(0.9, 5, 5)
diag(S) <- 1
mvrnorm(n = 10, rep(1, 5), Sigma = S)
```

³<http://forums.cirad.fr/logiciel-R/>

⁴<http://forums.cirad.fr/logiciel-R/viewtopic.php?f=1&t=7638>

```
Error in mvrnorm(n = 10, rep(1, 5), Sigma = S) :  
  impossible de trouver la fonction "mvrnorm"
```

On ne sais pas dans quel paquet se trouve la fonction `mvrnorm()`.

Exemple 1 reproductible:

```
library(MASS)  
S <- matrix(0.9, 5, 5)  
diag(S) <- 1  
mvrnorm(n = 10, rep(1, 5), Sigma = S)
```

Exemple 2 *non* reproductible:

```
z <- read.table("courbe.txt", header = TRUE, sep='\t')  
xyplot(y~x, data = z)  
Error in file(file, "rt") : impossible d'ouvrir la connexion  
De plus : Warning message:  
In file(file, "rt") :  
  impossible d'ouvrir le fichier 'courbe.txt' : No such file or directory
```

Le fichier `courbe.txt` n'est pas sur le disque. De plus, le paquet de la fonction `xyplot()` n'est pas précisé.

Exemple 2 reproductible:

```
library(lattice)  
z <- structure(list(x = 1:10, y = c(0, 0.693147180559945, 1.09861228866811,  
1.38629436111989, 1.6094379124341, 1.79175946922805, 1.94591014905531,  
2.07944154167984, 2.19722457733622, 2.30258509299405)), .Names = c("x",  
"y"), row.names = c(NA, -10L), class = "data.frame")  
xyplot(y~x, type = 'l', data = z)
```

Remarque : un code qui provoque une erreur est reproductible si l'erreur est l'objet de la question.

Exemple 3 reproductible. Question : « Je ne comprends pas cette erreur »

```
x <- 0  
if (x = 0) x <- NA  
Erreur : '=' inattendu(e) in "if (x ="
```

Ce code est reproductible car l'erreur produite est l'objet de la question.

Comment vérifier qu'un code est reproductible avant de le poster sur le forum ?
C'est très simple:

1. Vérifier que le code ne va lire aucun fichier de données sur disque ;
2. Sauver le code et quitter ;
3. Relancer  et copier-coller le code dans la fenêtre commande.

Si le code donne la sortie attendue, il est reproductible.

2.3 Comment insérer des données dans un message

CETTE section est directement adaptée des recommandations de François Bonnot du 2010-10-29⁵ sur le forum francophone des utilisateurs du logiciel .

DE nombreux messages échangés sur le forum nécessitent l'insertion d'un jeu de données permettant d'illustrer la question soumise. Afin d'alléger le contenu du message et de faciliter l'utilisation du jeu de données (et donc de maximiser les chances d'obtenir une réponse), il est vivement recommandé de respecter les principes suivants:

1. Réduire le jeu de données au minimum nécessaire à la résolution du problème. Dans la plupart des cas, un tableau d'une dizaine de lignes et quelques colonnes suffit à illustrer un problème de syntaxe. S'il s'agit de données réelles nécessitant un avis sur la pertinence d'une analyse, on peut dépasser cette limite, mais l'insertion de données dépassant un écran devrait être exceptionnelle (ne pas décourager le lecteur et penser aussi à notre espace disque). Au delà de cette limite, utiliser un lien externe.
2. Utiliser la fonction `dput()` pour insérer des données. Le lecteur dispose ainsi de l'image exacte des données, y compris leur classe et mode de stockage ce qui est souvent important.

Exemple. Soit le data.frame suivant:

```
df <- data.frame(x = letters[1:4], y = 1:4, z = log(1:4))
df
  x y      z
1 a 1 0.000000
2 b 2 0.6931472
3 c 3 1.0986123
4 d 4 1.3862944
```

Exécuter `dput()`:

```
dput(df)
structure(list(x = c("a", "b", "c", "d"), y = 1:4, z = c(0, 0.693147180559945,
1.09861228866811, 1.38629436111989)), class = "data.frame", row.names = c(NA,
-4L))
```

Editer ensuite cette sortie en ajoutant en tête le nom de l'objet et l'opérateur d'affectation, puis l'insérer dans le corps du message avec les balises « code » afin que le lecteur puisse charger directement les données:

```
df <- structure(list(x = structure(1:4, .Label = c("a", "b", "c", "d"),
), class = "factor"), y = 1:4, z = c(0, 0.693147180559945, 1.09861228866811,
1.38629436111989)), .Names = c("x", "y", "z"), row.names = c(NA,
-4L), class = "data.frame")
```

Afin de faciliter la lecture du message, on peut présenter le début du tableau en format lisible (voire l'insérer en entier s'il est très court) avant d'insérer la sortie de `dput()`. Exemple:

« Je dispose d'un data.frame de 10 lignes structuré ainsi: »

⁵<http://forums.cirad.fr/logiciel-R/viewtopic.php?f=1&t=3302>

```
  x y z
1 a 1 0.0000000
2 b 2 0.6931472
...
10 j 10 2.3025851
```

« et dont le contenu est: »

```
df <- structure(list(x = structure(1:4, .Label = c("a", "b", "c", "d"),
), class = "factor"), y = 1:4, z = c(0, 0.693147180559945, 1.09861228866811,
1.38629436111989)), .Names = c("x", "y", "z"), row.names = c(NA,
-4L), class = "data.frame")
```

2.4 Comment détailler son environnement de travail

La fonction `sessionInfo()` permet de lister automatiquement tout un tas de caractéristiques de votre environnement de travail tels que la version de  que vous utilisez, votre système d'exploitation, la version des bibliothèques BLAS et LAPACK pour les calculs matriciels, les options régionales et linguistiques et les versions des différents paquets  disponibles. Ces précisions peuvent dans certains cas de figure être utiles pour comprendre votre problème.

```
sessionInfo()
R version 4.1.1 (2021-08-10)
Platform: x86_64-apple-darwin17.0 (64-bit)
Running under: macOS Big Sur 10.16
Matrix products: default
BLAS: /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRblas.0.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRlapack.dylib

locale:
[1] fr_FR.UTF-8/fr_FR.UTF-8/fr_FR.UTF-8/C/fr_FR.UTF-8/fr_FR.UTF-8

attached base packages:
[1] stats graphics grDevices utils datasets methods base

other attached packages:
[1] lattice_0.20-44

loaded via a namespace (and not attached):
[1] compiler_4.1.1 tools_4.1.1 grid_4.1.1 fortunes_1.5-4
```

PRENONS un exemple concret. Supposons qu'en butinant sur la toile, à la recherche d'exemples d'utilisation de la commande `sort()` pour trier des chaînes de caractères dans l'ordre du dictionnaire, vous soyez tombé sur l'exemple de la commande  et de son résultat suivant :

```
sort(c(letters, "aa"))
[1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p"
[17] "q" "r" "s" "t" "u" "v" "w" "x" "y" "z" "aa"
```

VOUS essayez alors de reproduire ce résultat sur votre propre ordinateur, mais rien n'y fait, vous obtenez toujours un résultat différent. Vous avez un problème de *reproductibilité*, le "aa" qui se trouve en dernière position dans l'exemple se trouve en seconde position chez vous :

```
sort(c(letters, "aa"))
[1] "a" "aa" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o"
[17] "p" "q" "r" "s" "t" "u" "v" "w" "x" "y" "z"
```

Ce genre de problème peut être particulièrement frustrant parce que l'on ne voit vraiment pas la différence entre l'exemple glané sur internet et sa propre version. C'est ici que `sessionInfo()` peut être utile en montrant que l'exemple a été exécuté avec des options régionales et linguistiques danoises alors que les vôtres sont françaises. Or, il appert que dans l'ordre linguistique danois `aa` est considéré comme une lettre en soi et doit être placé après le `z`. D'où la différence, parfaitement logique, entre les deux résultats.

2.5 L'étiquette [Résolu]

SI vous posez correctement votre question sur le *bon* forum de discussion, il est probable que l'on vous apporte des éléments de solution. Par exemple, lors de la rédaction de ce document, je n'arrivais pas à produire simplement un document PDF utilisant la classe L^AT_EX « article » (on en reparlera dans la section 3.8 page 27). J'ai donc posé la question sur un forum de discussion⁶ et moins de 33 minutes plus tard j'avais déjà la solution. Dans ce cas de figure, il est d'usage de modifier le titre initial du fil de discussion pour lui ajouter une étiquette [Résolu] :



L'AVANTAGE de procéder ainsi est que dès la lecture du sujet on sait que l'on y trouvera une solution sans même avoir besoin de dérouler tout le fil. Pensez aux suivants qui auront le même problème que vous.

3 Approche professionnelle : utilisation de `kintr` et `RMarkdonw`

3.1 Qu'est ce que `kintr` ?

Le paquet `Rkintr` [102, 101, 105] (*tricoter*) est le descendant de la commande `Sweave()` (*coudre*) : il permet d'écrire des rapports où les résultats des analyses sont directement intégrés dans le document final. D'où cette notion de « tricottage » entre les résultats des calculs et votre texte pour produire le rapport final.

3.2 Qu'est ce que `RMarkdown` ?

C'EST un langage de balisage léger [107, 108, 2] permettant de séparer la forme du fond dans votre document. L'idée est de vous concentrer sur le fond du document et de laisser la mise en forme se faire automatiquement. Par exemple pour énumérer les éléments d'une liste finie on écrit :

⁶<http://forums.cirad.fr/logiciel-R/viewtopic.php?f=3&t=10477>

Les trois valeurs de vérité dans le logiciel R sont :

1. TRUE
2. FALSE
3. NA

C'EST ensuite `knitr` qui s'occupera de la mise en forme dans le document final, par exemple comme ceci :

Les trois valeurs de vérité dans le logiciel R sont :

1. TRUE
2. FALSE
3. NA

CE langage de balisage est dit « léger » dans le sens où le texte non formaté reste assez lisible pour un être humain par rapport à d'autres langages de balisage. Par exemple, avec `LATEX` on aurait écrit :

Les trois valeurs de vérité dans le logiciel R sont :

```
\begin{enumerate}
  \item TRUE
  \item FALSE
  \item NA
\end{enumerate}
```

L'AVANTAGE de `RMarkdown` est donc sa bonne lisibilité. Son principal inconvénient est une plus grande fragilité. Si vous oubliez par exemple de sauter une ligne entre le chapeau introductif et le début de l'énumération :

Les trois valeurs de vérité dans le logiciel R sont :

1. TRUE
2. FALSE
3. NA

cela ne fonctionnera plus et vous aurez un résultat du type :

Les trois valeurs de vérité dans le logiciel R sont : 1. TRUE 2. FALSE 3. NA

LES éléments de mise en forme de `RMarkdown` s'inspirent de ce qui se pratique dans les courriels, ou plus généralement dans les fichiers textes bruts, par exemple :

```
knitr/courriel.txt
-----
Bonjour,

je vous ai déjà dit qu'il est fort *impoli* que d'envoyer des
courriels au format HTML. J'ai désactivé cette fonctionnalité
car elle est utilisée par des polluposteurs pour vérifier
l'efficacité de leurs **pourriels**.

Cordialement,

Net I. Kett
```

Le fichier RMarkdown :

```
knitr/courriel.Rmd
---
output: html_document
---
Bonjour,

je vous ai déjà dit qu'il est fort *impoli* que d'envoyer des
courriels au format HTML. J'ai désactivé cette fonctionnalité
car elle est utilisée par des polluposteurs pour vérifier
l'efficacité de leurs **pourriels**.

Cordialement,

Net I. Kett
```

donnera :

```
Bonjour,

je vous ai déjà dit qu'il est fort impoli que d'envoyer des courriels au format HTML. J'ai désactivé cette
fonctionnalité car elle est utilisée par des polluposteurs pour vérifier l'efficacité de leurs pourriels.

Cordialement,

Net I. Kett
```

L'ITALIQUE est utilisé pour mettre *de l'emphase* sur une partie du texte. Les caractères **gras** ne s'emploient pas [5] dans les documents destinés à la lecture mais dans des documents destinés à être parcourus (comme un dictionnaire) parce qu'ils attirent trop l'œil, comme on peut s'en rendre compte sur ce paragraphe. Il faut éviter d'utiliser l'italique pour de trop longs fragments de texte : des études de suivi des mouvements de l'œil lors de la lecture montrent qu'elle est difficile pour des textes en italique, en particulier pour des polices de caractères sans empattements [82, 83]. Pour citer Jacques ANDRÉ [5] : « [a]buser de l'italique reviendrait à perdre son côté “mise en vedette” ». Pour un exemple extrême voir la figure 2 page 16, mais voici une première illustration :

Il faut éviter d'utiliser l'italique pour de trop longs fragments de texte : des études de suivi des mouvements de l'œil lors de la lecture montrent qu'elle est difficile pour des textes en italique, en particulier pour des polices de caractères sans empattements [82, 83]. Pour citer Jacques ANDRÉ [5]: « [a]buser de l'italique reviendrait à perdre son côté “mise en vedette” ». Pour un exemple extrême voir la figure 2 page 16. QED

EXCLUSIONS DE GARANTIE

Sauf accord écrit, la Garantie limitée ne s'applique pas aux circonstances suivantes : (a) l'usure normale (y compris, sans s'y limiter, l'usure des batteries le cas échéant) ; (b) les défauts causés par une manipulation ou une utilisation brutale ou inappropriée, les dommages dus à un accident, une mauvaise utilisation, la négligence, le feu, l'eau, la foudre ou d'autres phénomènes naturels ; (c) si le Produit contient une batterie, un court-circuit de batterie, si les fermetures du boîtier de la batterie ou les cellules sont endommagées ou présentent des signes de manipulation ou si la batterie a été utilisée dans des appareils autres que ceux auxquels elle est destinée ; (d) toute non-conformité avec les instructions d'utilisation du Produit ; (e) les dommages infligés de façon volontaire ou intentionnelle, la négligence ou l'imprudence ; (f) l'utilisation de pièces de rechange, d'une solution de nettoyage non autorisée le cas échéant ou d'autres éléments de remplacement (y compris les consommables) non fournis ni recommandés par nous ; (g) toute altération ou modification du Produit effectuée par vous ou un tiers non approuvé par nous ; (h) tout transport du Produit sans emballage adéquat ; (i) des causes extrêmes ou externes indépendantes de notre contrôle raisonnable, y compris, mais sans s'y limiter, les coupures, fluctuations ou interruptions de courant électrique, des services du fournisseur d'accès à Internet ou des réseaux sans fil ; (j) un signal sans fil faible et/ou discontinu dans votre maison ; (k) tout Produit acheté auprès de vendeurs/revendeurs non agréés.

de vendeurs/revendeurs non agréés. (avec une police de 5 points pica)

de vendeurs/revendeurs non agréés. (avec une police de 5 points Didot)

Figure 1: De l'utilisation de caractères minuscules. Exemple ici en taille réelle des exclusions de garantie listées dans la notice accompagnant l'achat d'un robot ménager en France en 2020. Dans les documents composés avec L^AT_EX comme ici, le point typographique par défaut (point pica) vaut $\frac{2,54}{72,27} \approx 0,03514598$ cm, soit légèrement moins que le point Didot traditionnel en typographie française (0,03760650 cm). On voit que la police utilisée pour la notice fait environ 5 points pica, ce qui est très petit. Pour donner un élément de comparaison, le code de la consommation dispose en son article R312-10 dans la version consolidée au 11 janvier 2021 que « [l]e contrat de crédit prévu à l'article L. 312-28 est rédigé en caractères dont la hauteur ne peut être inférieure à celle du corps huit ». On ne sait pas s'il s'agit d'un corps de 8 points pica ou de 8 points Didot, mais à 5 points pica on est bien en dessous du minimum requis.

SURVEILLANCE VIROLOGIQUE

La surveillance virologique basée sur les laboratoires permet de déterminer et suivre l'évolution, dans le temps et par région ou département, des taux d'incidence (nombre de personnes testées positives pour le SARS-CoV-2 rapporté à la population), des taux de positivité (nombre de personnes testées positives pour le SARS-CoV-2 rapporté au nombre de personnes testées) et des taux de dépistage (nombre de personnes dépistées rapporté à la population).

Jusqu'au 12 mai 2020, cette surveillance s'est appuyée sur les données non exhaustives transmises à Santé publique France par le réseau 3 Labo (Cerba, Eurofins-Biomnis, Inovie) et par les laboratoires hospitaliers. Depuis le 13 mai, elle repose sur le système SI-DEP (système d'information de dépistage), dont la montée en charge a été progressive.

SI-DEP vise au suivi exhaustif de toutes les personnes testées en France pour le diagnostic et le dépistage de la COVID-19 dans les laboratoires de ville, les laboratoires hospitaliers et par les autres professionnels de santé.

Sont actuellement pris en compte dans les indicateurs SI-DEP les tests RT-PCR, les tests antigéniques réalisés en laboratoire (TDR, depuis le 17 octobre) ainsi que ceux réalisés hors laboratoire (TROD, depuis le 16 novembre). Dans cette partie, les tests RT-LAMP sont assimilés à des tests RT-PCR.

Lorsque le code postal de la personne testée est disponible, les données sont attribuées au département et région de résidence ; lorsqu'il ne l'est pas, les données ne sont prises en compte que pour les analyses nationales (région et département non disponibles).

Cas confirmé de COVID-19 : une personne présentant une infection par SARS-CoV-2 confirmée par test RT-PCR ou test antigénique, que cette personne soit symptomatique ou asymptomatique (voir [définition de cas](#)).

Figure 2: Du brouillage typographique d'une communication institutionnelle. Exemple ici en taille réelle d'une copie d'écran du haut de la page 6 du point épidémiologique hebdomadaire¹ du 7 janvier 2021 sur la covid 19 publié par « Santé publique France » sur son site. Les fautes typographiques sont dans l'ordre de gravité croissant :

- 1° l'utilisation à mauvais escient du gras² ;
- 2° l'utilisation d'une police sans empattements en italique prouvée expérimentalement d'être de piètre lisibilité³ ;
- 3° *last, but not least*, un nombre de caractères par lignes beaucoup trop grand : les lignes complètes font ici de 108 à 122 caractères⁴.

Point n'est besoin de long discours : il suffit de comparer la fluidité de lecture de la présente légende avec le difficile déchiffrement de cet exemple. Les réglages par défaut du logiciel de composition L^AT_EX donnent des résultats d'excellente qualité typographique, il est donc prudent de ne pas y toucher.

¹ Fichier COVID19_PE_20210107.pdf à <https://www.santepubliquefrance.fr/>

² Pour une explication détaillée on consultera les pages 21 à 29 des *Petites leçons de typographie* de Jacques André [5].

³ Il s'agit d'expériences de suivi en temps réel des mouvements des yeux pendant la lecture qui permettent d'apprécier objectivement la fluidité de cette dernière [82, 83].

⁴ Il est recommandé de rester dans une plage allant de 45 à 75 avec 66 caractères par ligne comme optimum souvent utilisé ([14] cité par [78]).

Au-delà, un.e étudiant.e ne pourra être admis.e à s’inscrire une nouvelle fois dans un de ces niveaux que sur décision du Président de l’université (ou de son délégataire), après avis favorable de la commission *ad hoc* qui sera nommée par le.la directeur-trice de la composante responsable de la formation et composée de :

- le.la directeur.trice de la composante ou son.sa représentant.e ;
- un.e directeur.trice des études ou son.sa représentant.e ;
- un.e représentant.e du service SOIE;
- le.la responsable de la formation ou son.sa représentant.e ;
- d’un.e étudiant.e de l’année supérieure mais pas nécessairement de la même mention, élu.e de la CFVU, ou à défaut, en cas d’impossibilité, un.e élu.e du conseil de composante, ou à défaut, en cas d’impossibilité, un.e élu.e du conseil de département.

Figure 3: L’écriture pointée dite « inclusive¹ » messied particulièrement à la lisibilité tant typographique que sémantique d’un texte. Copie d’écran d’une partie du document du point II de la partie A de la CFVU² exceptionnelle du 12 juillet 2017. Cette écriture employée naguère est tombée en disgrâce le 21 novembre 2017³. Le même passage composé avec L^AT_EX de façon plus canonique donnerait :

AU-DELÀ, un étudiant ne pourra être admis à s’inscrire une nouvelle fois dans un de ces niveaux que sur décision du président de l’université (ou de son délégataire), après avis favorable de la commission *ad hoc* qui sera nommée par le directeur de la composante responsable de la formation et composée de :

- 1° le directeur de la composante ou son représentant ;
- 2° un directeur des études ou son représentant ;
- 3° un représentant du service SOIE;
- 4° le responsable de la formation ou son représentant ;
- 5° d’un étudiant de l’année supérieure mais pas nécessairement de la même mention, élu de la CFVU, ou à défaut, en cas d’impossibilité, un élu du conseil de composante, ou à défaut, en cas d’impossibilité, un élu du conseil de département.

¹ C’est le terme employé par ses zéloteurs mais elle plus perçue comme étant discriminante par les aveugles et handicapés visuels, les dyslexiques, les lecteurs francophones dont ce n’est pas la langue maternelle et les personnes de sexe indéterminé.

² Acronyme de la « Commission de la Formation et de la Vie Universitaire » de l’université Claude Bernard - Lyon I.

³ Circulaire du 21 novembre 2017 relative aux règles de féminisation et de rédaction des textes publiés au Journal officiel de la République française.

POUR mettre en exposant il faut écrire x^2 , pour mettre en indice il faut écrire x_1 mais notez que le rendu de x^{2_1} ou de x_1^2 ne sera pas bon et qu'il faut passer par une formule $\text{\LaTeX } \$x_1^2\$$ pour obtenir x_1^2 . D'une façon générale, il faut *toujours* passer par \LaTeX pour les formules mathématiques. Les éléments de code sont encadrés par des accents graves, par exemple `\mafonction()`. Une note de bas de page s'obtient par `^[texte]`⁷. Les liens hypertexte sont rendus par `[texte](lien)` et les images insérées par `![texte](lien)`.

LA syntaxe pour rendre les petites capitales, indispensables en typographie française, n'est pas des plus compacte parce qu'elle passe par une balise HTML `span`. Dans l'exemple suivant la structuration en vers a été préservée dans le document final en terminant chaque ligne par deux espaces. On en profite ici pour illustrer la notation HTML pour les commentaires `<!-- commentaire qui ne figurera pas dans le document final -->`. Dans RStudio il y a un raccourci clavier qui permet de (dé)commenter facilement une ligne.

```
----- knitr/Corneille.Rmd -----
---
output: pdf_document
---

<span style="font-variant:small-caps;">Polyeucte.</span>
Vous me connaissez mal : la même ardeur me brûle
Et le désir s'accroît quand l'effet se recule :
Ces pleurs, que je regarde avec un œil d'époux,
Me laissent dans le cœur aussi chrétien que vous ;

<span style="font-variant:small-caps;">Polyeucte.</span>
<!-- Vous me connaissez mal : la même ardeur me brûle -->
Et le désir s'accroît quand l'effet se recule :
<!-- Ces pleurs, que je regarde avec un œil d'époux, -->
<!-- Me laissent dans le cœur aussi chrétien que vous ; -->
```

Donnera, même dans un document PDF :

```
POLYEUCTE. Vous me connaissez mal : la même ardeur me brûle
Et le désir s'accroît quand l'effet se recule :
Ces pleurs, que je regarde avec un œil d'époux,
Me laissent dans le cœur aussi chrétien que vous ;

POLYEUCTE. Et le désir s'accroît quand l'effet se recule :
```

3.3 Qu'est-ce que pandoc ?

C'EST la cheville ouvrière qui travaille pour convertir votre document `RMarkdown` en un document final au format désiré. Vous ne l'utiliserez pas directement mais il est bien là, dans l'ombre, à faire sa magie. C'est lui qui par exemple traduira pour vous les notations obsolètes pour les formules $\text{\LaTeX } \$\dots\$$ (formule dans le texte) et $\text{\LaTeX } \$\dots\$\$$ (formule détachée du texte) en `\(\dots\)` et `\[\dots\]`, respectivement.

⁷Pour un rapport au format PDF il s'agit de véritables notes de bas de page, pour un rapport au format HTML qui n'est pas structuré en pages on parle plutôt de notes de fin de document.

3.4 Pré-requis : installer un paquet

NOUS supposons ici que  et RStudio ont déjà été installés sur votre ordinateur, c'est en principe déjà le cas pour les machines en salle de TP. Si vous travaillez avec votre machine personnelle et que vous hésitez à vous lancer dans l'aventure, nous vous invitons à regarder une vidéo⁸ en français qui devrait vous convaincre que ce n'est pas si compliqué que ça.

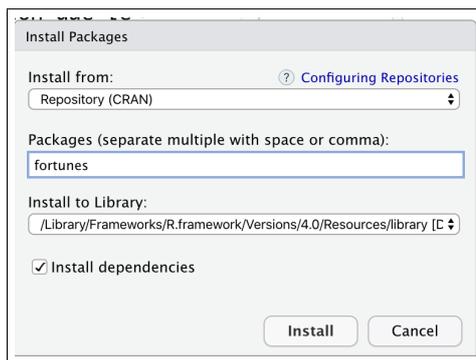
LES paquets  (*package* en anglais) sont des extensions qui ajoutent de nouvelles fonctionnalités. Il y a un nombre considérable de paquets disponibles (18266 lors de la dernière compilation de ce document). Il est donc intéressant que d'être capable de profiter de tous ces développements. Cela peut être plus ou moins compliqué selon que le paquet nécessite d'être compilé (4348 paquets) ou non (13917 paquets).



Paquet vient de l'anglais *package* et évoque un paquet-cadeau.

3.4.1 Paquets ne nécessitant pas d'être compilés

C'EST le cas le plus courant. Il s'agit de paquets ne nécessitant que du code  pour fonctionner. Nous allons utiliser le paquet `fortunes` [109] pour illustrer l'installation et l'utilisation d'un paquet. Dans RStudio sélectionnez *Tools* → *Install Packages* puis renseignez le paquet à installer puis cliquez sur le bouton *Install* :



UNE fois l'installation terminée nous allons essayer d'utiliser le paquet. Le paquet `fortunes` définit essentiellement la fonction `fortune()` qui permet de récupérer des éléments du folklore geek de . Essayez de rentrer `fortune("Lobry")` dans la console :

```
fortune("Lobry")
Error in fortune("Lobry") : impossible de trouver la fonction "fortune"
```

ÇA ne marche visiblement pas. Ce qu'il faut comprendre ici c'est que ce n'est pas parce qu'un paquet est *installé* qu'il directement *utilisable*. Pour utiliser la fonction `fortune()` du paquet `fortunes` il faut dire à  où la trouver. Il y a deux possibilités pour ce faire, la première est d'utiliser la notation `::` pour expliciter dans quel paquet se trouve la fonction invoquée :

```
fortunes::fortune("Lobry")
It seems that you are facing a very serious fortune(122) problem.
-- Jean R. Lobry (in reply to a user that quickly needed help for his PhD thesis,
   without properly checking the documentation)
   R-help (May 2006)
```

⁸https://www.youtube.com/watch?v=sav3Mbe0_DM

LA seconde possibilité est d'utiliser la fonction `library()` pour dire à  d'explorer systématiquement un paquet donné. Il n'y a alors plus besoin d'utiliser la notation `::` pour faire appel à une fonction de ce paquet :

```
library("fortunes")
fortune(122)
```

```
I wish to perform brain surgery this afternoon at 4pm and don't know where to start.
My background is the history of great statistician sports legends but I am willing
to learn. I know there are courses and numerous books on brain surgery but I don't
have the time for those. Please direct me to the appropriate HowTos, and be on
standby for solving any problem I may encounter while in the operating room. Some of
you might ask for specifics of the case, but that would require my following the
posting guide and spending even more time than I am already taking to write this
note.
```

```
-- I. Ben Fooled (aka Frank Harrell)
R-help (April 1, 2005)
```

LA seconde possibilité peut sembler plus légère d'utilisation. Elle présente cependant un inconvénient majeur : si deux paquets ont une fonction ayant le même nom, vous ne saurez plus laquelle vous invoquez, ce qui peut être la source d'une confusion certaine. Il est plus prudent d'utiliser la première possibilité.

UNE situation tout à fait analogue va se présenter lorsque vous manipulerez des `data.frame` dans . En statistique, une convention universelle, respectée par tous les logiciels, veut que les unités statistiques soient en ligne et les variables en colonne. Les `data.frame` respectent donc cette convention. Pour accéder aux valeurs d'une colonne, vous avez deux possibilités. La première est d'utiliser l'opérateur `$` :

```
data(women)
women$height
[1] 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72
```

LA seconde possibilité est d'utiliser la fonction `attach()` pour dire à  d'explorer systématiquement un jeu de données. Il n'y a alors plus besoin d'utiliser la notation `$` pour récupérer une colonne du `data.frame` :

```
attach(women)
height
[1] 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72
```

ICI encore, la seconde possibilité peut sembler plus légère d'utilisation. Elle présente cependant le même inconvénient majeur : si deux `data.frame` ont une colonne ayant le même nom, vous ne saurez plus à laquelle vous vous référez. Il est plus prudent d'utiliser la première possibilité. Si vous tenez absolument à ne pas utiliser l'opérateur `$` il y a moyen de le faire temporairement avec la fonction `with()` :

```
with(women, height)
[1] 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72
```

3.4.2 Paquets nécessitant d'être compilés

CE n'est pas le cas le plus fréquent puisque ces paquets ne représentent que de l'ordre du quart des paquets disponibles. Selon votre système d'exploitation, l'installation se fera soit par l'utilisation d'un paquet pré-compilé pour vous, soit par la compilation à la volée du paquet. Dans ce dernier cas

vous devez avoir les outils de compilation disponibles. Pour tester si vous êtes capables d'utiliser un paquet nécessitant d'être compilé, essayez d'installer le paquet `seqinr` [20] puis entrez dans la console :

```
seqinr::rot13("Ergheaf gur EBG-13 pvcurevat bs n fgevat")  
[1] "Returns the ROT-13 ciphering of a string"
```

3.4.3 Paquets nécessitant un programme externe

POUR être complet il convient de signaler qu'il existe également des paquets (environ 10 %) qui ont besoin de programmes externes pour fonctionner, par exemple le paquet `ORacle` [73] a besoin qu'un client ou pilote Oracle soit installé. Dans ce cas de figure vous aurez probablement besoin d'avoir les droits d'administrateur pour pouvoir installer les programmes nécessaires sur votre ordinateur.

3.5 Pré-requis : installer \LaTeX

3.5.1 La distribution Tiny \TeX

VOUS avez besoin d'avoir les outils nécessaires pour transformer un fichier source \LaTeX en un document PDF. Il en existe de nombreuses distributions classiques telles que `MiKTeX`, `MacTeX` et `TeX Live`, mais pour les utilisateurs de `RMarkdown` il est conseillé d'installer le paquet `tinytex` [104, 106] qui installera la distribution Tiny \TeX .

LA distribution Tiny \TeX est une version ultra épurée de `TeX Live`. Comme `TeX` a été créé il y a plus de 40 ans, un écosystème très riche et foisonnant s'est construit autour de lui et une installation complète est gourmande en ressources. L'idée derrière `tinytex` c'est de n'installer que le strict minimum et que si un paquet \LaTeX vient à manquer, on l'installera à la volée. Un autre avantage de Tiny \TeX est qu'il n'y a pas besoin d'avoir de compte administrateur pour l'installer, ce qui est bien pratique pour des étudiants en salle de TP. Une fois `tinytex` installé, l'installation de Tiny \TeX se fait simplement par :

```
tinytex::install_tinytex()  
# Pour désinstaller : tinytex::uninstall_tinytex()
```

QUITTEZ ensuite `RStudio` puis relancez le. Lorsque vous allez essayer de produire un document PDF il se peut très bien que vous obteniez d'obscurs messages d'erreur du type :

```
Running pdflatex on zefplock.tex...failed
```

```
Error: zefplock.tex:24: LaTeX Error: File `fancyhdr.sty' not found.  
Error: zefplock.tex: ==> Fatal error occurred, no output PDF file produced!
```

DANS ce cas, pas de panique. Si votre fichier de départ s'appelle `zefplock.Rmd`, un fichier `zefplock.log` décrivant les erreurs va être généré dans le même dossier. Il suffit de demander à l'analyser avec :

```
tinytex::parse_install("zefplock.log")
```

SI vous avez un message d'erreur comme quoi il est impossible d'ouvrir le fichier `zefplock.log` c'est que votre dossier de travail n'est pas le bon. Une première solution est d'ouvrir votre fichier `zefplock.Rmd` puis de faire *Session* → *Set Working Directory* → *To Source File Location*. Une deuxième solution est d'aller chercher votre fichier `zefplock.log` à la main avec :

```
tinytex::parse_install(file.choose())
```

3.5.2 Un petit test

C'EST maintenant le moment de vérité. Vérifiez que vous êtes capable de compiler le petit fichier (`mini.tex`) source L^AT_EX suivant :

```
knitr/minitex.tex
\documentclass[french,a4paper]{article}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{lmodern}
\usepackage{babel}
\begin{document}
\begin{Huge}Bonjour, \delta mon\delta! \end{Huge}
\end{document}
```

Pour ce faire suivez les étapes suivantes :

1. Créez-vous un dossier de travail, par exemple `testlatex`.
2. Allez sur le site <https://pbil.univ-lyon1.fr/R/donnees/knitr/> pour enregistrer une copie du fichier `minitex.tex` dans votre dossier de travail.
3. Dans RStudio, ouvrez le fichier `minitex.tex` présent dans votre dossier de travail `testlatex`.
4. Sélectionnez *Session* → *Set Working Directory* → *To Source File Location*
5. Cliquez sur « Compile PDF ».
6. Si vous avez des messages d'erreur entrez `tinytex::parse_install("minitex.log")` dans la console  puis retournez à l'étape précédente autant de fois que nécessaire.

VOUS devez obtenir un document PDF d'une page au format A4, numérotée 1 en bas, et sur le haut de laquelle est écrit en gros caractères :

Bonjour, Ô monde !

Ce petit test vous permet de vérifier deux choses :

1. Vous êtes capable de compiler un fichier source L^AT_EX qui contient des caractères accentués, comme ici le caractère « Ô ».
2. Le caractère grec delta minuscule, δ , est plaisant à lire. En effet, par rapport à la version initiale de 1986, le rendu du caractère δ a été fortement amélioré en 1992⁹. Si le rendu du caractère δ n'est pas bon (*cf.* ci-contre) c'est sans doute que votre version de L^AT_EX est légèrement obsolète et qu'il faut absolument faire une mise à jour.

⁹Voir <http://www-cs-faculty.stanford.edu/~uno/cm.html>



δ laid



δ beau

3.5.3 Mise à jour

LA distribution `tinyTeX` est mise à jour avec une périodicité annuelle. Si votre version est trop vieille vous aurez un message du type :

```
tlmgr: Remote repository is newer than local (2017 < 2018)
```

POUR mettre à jour votre distribution `tinyTeX` sans avoir à ré-installer tous les paquets `LATEX` que vous aviez déjà installés il suffit de faire :

```
tinytex::reinstall_tinytex()
```

POUR installer un paquet `LATEX` particulier on utilise la commande `tlmgr_install()`, par exemple pour avoir les règles de césure à la française :

```
tinytex::tlmgr_install("hyphen-french")
```

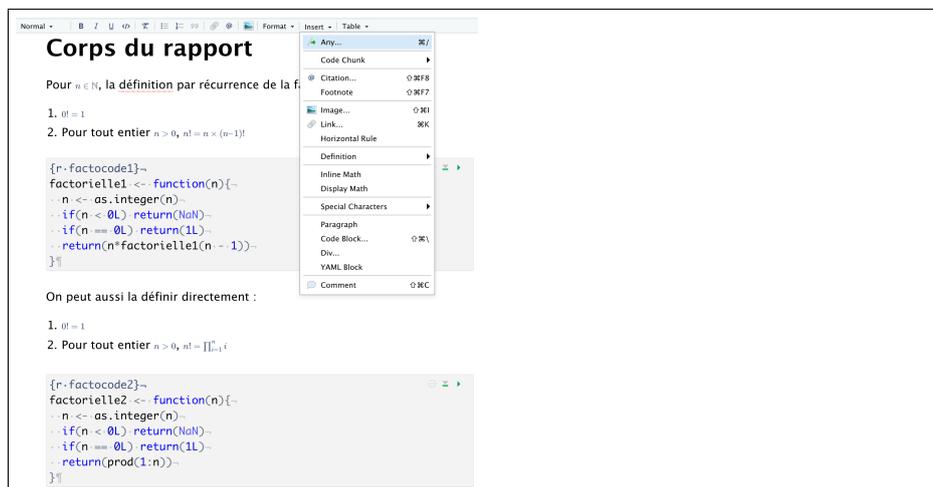
3.5.4 Pré-visualisation

COMME nous allons le voir dans la section suivante, les formules au format `LATEX` peuvent être directement utilisées dans un fichier `RMarkdown`. Un avantage à utiliser `RStudio` est qu'un aperçu du rendu final vous est immédiatement donné dans un encart quand le curseur est dans la formule :



3.5.5 Le mode visuel

À partir de la version 1.4 de `RStudio` il est possible de basculer dans un mode dit visuel qui ressemble plus à un traitement de texte traditionnel :



C'EST un peu l'équivalent de `LyX` pour `LATEX`, et peut être rassurant pour un étudiant qui hésiterait à abandonner son traitement de texte favori pour rédiger un rapport. Notons que si pour une raison ou une autre il n'était pas satisfait de sa migration, il aurait toujours la sécurité de pouvoir ré-exporter le travail déjà effectué vers son traitement de texte favori.

3.6 Un exemple RMarkdow minimaliste

```
knitr/mini.Rmd
---
output: html_document
---
# Bonjour, Ô mon$\delta$e !

Pour vérifier que $\LaTeX$ n'est pas trop obsolète...
```

1. Créez-vous un dossier de travail, par exemple `testrmd`.
2. Allez sur le site vu précédemment pour enregistrer une copie du fichier `mini.Rmd` dans votre dossier de travail. Notez que les fichiers RMarkdown ont par convention l'extension `*.Rmd`.
3. Dans RStudio, ouvrez le fichier `mini.Rmd` présent dans votre dossier de travail.
4. Sélectionnez *Session* → *Set Working Directory* → *To Source File Location*
5. Cliquez sur « Knit to HTML ».

Une nouvelle fenêtre devrait apparaître avec le résultat suivant :

```
Bonjour, Ô monde !
Pour vérifier que  $\LaTeX$  n'est pas trop obsolète...
```

La section comprise entre les trois tirets (`---`) contient les métadonnées du document final, ici `output: html_document` signifie que l'on veut produire un document au format `html`, un format adapté à une publication sur la toile. C'est à ce niveau que l'on contrôle la forme du document.

Le caractère `#` suivi d'une espace sert à indiquer un titre de niveau 1. Un titre de niveau 2 serait indiqué avec `##`, de niveau 3 avec `###`, etc. Le titre lui même doit être sur une seule ligne. Le caractère `$` sert à encadrer des formules mathématiques au format `LATEX`, le standard pour l'édition scientifique. Nous verrons quelques exemples simples plus tard.

DANS le jargon RMarkdownien la section comprise entre les trois tirets s'appelle le `YAML`, ce que l'on pourrait traduire en français de façon plus descriptive par « l'en-tête » ou bien par attraction de `LATEX` par le « préambule » du document. Ces dénominations sont un peu trompeuses dans le sens où le `YAML` peut être dispersé dans tout le document, c'est `knitr` qui s'occupera de recoller les morceaux. On peut avoir ainsi des constructions de la forme :

```
knitr/foot.Rmd
---
output: html_document
---
```{r}
gagnant <- ifelse(runif(1) < 0.5, "la France", "l'Allemagne")
```

Et c'est donc finalement `r gagnant` qui emporte cette dernière coupe
du monde de football...
```

```
---
title: Victoire de `r gagnant` !!!
---
```

Si vous compilez ce document vous aurez dans 50 % des cas :

Victoire de l'Allemagne !!!

```
gagnant <- ifelse(runif(1) < 0.5, "la France", "l'Allemagne")
```

Et c'est donc finalement l'Allemagne qui emporte cette dernière coupe du monde de football...

et dans 50 % des cas :

Victoire de la France !!!

```
gagnant <- ifelse(runif(1) < 0.5, "la France", "l'Allemagne")
```

Et c'est donc finalement la France qui emporte cette dernière coupe du monde de football...

AUTREMENT dit, on a tiré à pile ou face dans le corps du document avec la ligne de code `R runif(1) < 0.5` qui était le gagnant et on a utilisé ce résultat pour construire le titre du document. Cet exemple peut sembler un peu artificiel, mais imaginez maintenant que votre rapport soit destiné à être mis à jour périodiquement en fonction de données actualisées (*e.g.* un bulletin météo, un point de situation épidémiologique, le suivi d'une action cotée en bourse). Vous pouvez ainsi analyser de façon fine les données dans le corps de votre rapport et construire automatiquement un titre qui résume la tendance générale. Vous pouvez avoir dès maintenant une intuition de la différence de puissance qu'il y a entre maîtriser `RMarkdown` et tapoter dans un simple logiciel de traitement de texte.

DANS le même ordre d'idée, il est possible de produire des rapports *paramétrés*, c'est à dire dont le rendu final va dépendre de paramètres choisis au moment de la demande de mise en forme du document. Vous pouvez imaginer par exemple d'avoir à produire un rapport se restreignant à une certaine région du monde, à une certaine plage temporelle ou bien à un certain type de produits. Avec un seul document `RMarkdown` vous pourrez ainsi produire une foultitude de rapports différents. Pour donner un exemple très simple prenons la célèbre lettre dite de George SAND à Alfred DE MUSSET dont le contenu érotique n'est dévoilé qu'en lisant une ligne sur deux. Vous pouvez introduire un paramètre booléen contrôlant si on affiche une ligne sur deux ou non dans le document final.

knitr/Sand.Rmd

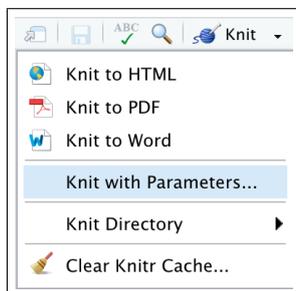
```
---
output: html_document
params:
  adusum:
    label: "Ad usum Delphini"
    value: TRUE
---

```{r echo=FALSE,results="asis"}
lgs <- c(
 "Je suis émue de vous dire que j'ai",
 "bien compris l'autre soir que vous aviez",
 "toujours une envie folle de me faire",
 "danser. Je garde le souvenir de votre",
 "baiser et je voudrais bien que ce soit",
 "là une preuve que je puisse être aimée",
 "par vous. Je suis prête à vous montrer mon",
 "affection toute désintéressée et sans cal-",
 "cul, et si vous voulez me voir aussi",

```

```
"vous dévoiler sans artifice mon âme",
"toute nue, daignez me faire une visite.",
"Nous causerons en amis, franchement.",
"Je vous prouverai que je suis la femme",
"sincère et capable de vous offrir l'affection",
"la plus profonde comme la plus étroite",
"en amitié, en un mot, la meilleure épouse",
"que vous puissiez rêver puisque votre",
"âme est libre. Pensez que la solitude où j'ha-",
"bite est bien longue, bien dure et souvent",
"difficile. Ainsi en y songeant j'ai l'âme",
"grosse. Accourrez donc vite et venez me la",
"faire oublier. A l'amour que je veux vous sou-",
"mettre. ")
lgs <- paste(lgs, " \n")
if(params$adusum) cat(lgs) else cat(lgs[c(TRUE, FALSE)])
...
```

Vous pouvez sélectionner le menu « Knit with parameters » :



pour obtenir la fenêtre de dialogue suivante :



qui vous permet de contrôler si vous voulez produire un rapport *ad usum Delphini* ou non.

### 3.7 Dossier de travail et reproductibilité

Le dossier de travail (*Working directory*) sert à regrouper en un même endroit tous les éléments nécessaires à la reproduction des résultats de votre rapport. C'est ce dossier que vous donnerez à ceux qui voudraient répéter vos analyses. Vous n'êtes pas obligé de laisser tout à plat dans ce dossier de travail, vous pouvez par exemple créer un sous-dossier `data` pour regrouper les jeux de données utilisés, un sous-dossier `figs` pour regrouper vos images, etc.

Pour faciliter la reproductibilité de vos résultats il est important de ne jamais utiliser de chemin *absolu* mais toujours un chemin *relatif*. Supposons que vous ayez un jeu de données `zefplock.csv` dans votre sous-dossier `data`. Un chemin absolu est quelque chose qui ressemble à

```
/Users/lobry/encours/testrmd/data/zefplock.csv
```

et qui est spécifique de votre machine : il n'existera pas sur l'ordinateur d'une autre personne et sera un facteur de blocage pour reproduire vos résultats. Un

chemin relatif sera simplement ici `data/zefplock.csv`, il consiste à prendre comme point de départ le dossier de travail.

CERTAINS systèmes d'exploitation utilisent une barre oblique inversée (`\`) comme séparateur dans les chemins d'accès aux fichiers. On a alors des constructions du type :

```
U:\lobry\encours\testrmd\data\zefplock.csv
```

Vous n'aurez pas à vous occuper de cette subtilité. En effet, quelque soit le système d'exploitation sous-jacent, dans  les chemins d'accès sont toujours codés avec une barre oblique directe (`/`). C'est  qui s'occupera de faire la traduction qui va bien pour vous. Si par malheur vous étiez dans la nécessité *absolue* d'avoir à utiliser ce type de notation, il faut savoir que le caractère (`\`) est utilisé dans les chaînes de caractères comme caractère d'échappement pour représenter des choses particulières comme (`\n`) pour un retour à la ligne. Pour représenter le caractère (`\`) lui-même, il faut le doubler (`\\`) :

```
chemin <- "U:\\lobry\\encours\\testrmd\\data\\zefplock.csv"
cat(chemin)
U:\lobry\encours\testrmd\data\zefplock.csv
fortunes::fortune("backslash")
When in doubt, keep adding slashes until it works.
-- Joran Elias (on how to escape a backslash in R)
Stackoverflow (March 2015)
```

### 3.8 Séparation de la forme et du fond

POUR illustrer la séparation de la forme et du fond nous allons modifier notre exemple minimaliste précédent pour dire avec `theme: cerulean` que nous voulons changer de thème de présentation :

```
----- knitr/minicerulean.Rmd -----

output:
 html_document:
 theme: cerulean

Bonjour, Ô monδe !

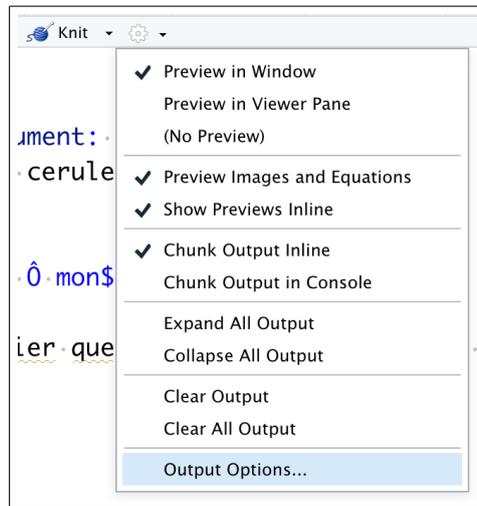
Pour vérifier que \LaTeX n'est pas trop obsolète...
```

En sélectionnant « Knit to HTML » vous devriez obtenir :

**Bonjour, Ô monde !**

Pour vérifier que  $\LaTeX$  n'est pas trop obsolète...

LA forme de votre document a changé : les titres de niveau 1 sont maintenant en couleur. Mais le fond de votre document n'a pas changé : il n'y a toujours qu'un titre de niveau 1. Vous pouvez explorer les différentes options de mise en forme avec le menu « Output options ».



POUR ce qui est de la forme, votre document doit avant tout être facilement *lisible*. Un défaut fréquent des documents produits avec des traitements de texte consiste à avoir des lignes trop longues. Comptez le nombre de caractères par ligne dans un livre publié par un éditeur professionnel. Vous constaterez qu'il y a environ 66 caractères par ligne. La raison est que la lecture devient difficile dès que l'on dépasse cette limite parce qu'il est alors pénible pour les yeux de se déplacer de la fin de la ligne courante au début de la ligne suivante. Démonstration :

*The proof of the pudding is in the reading.* Pour ce qui est de la forme, votre document doit avant tout être facilement *lisible*. Un défaut fréquent des documents produits avec des traitements de texte consiste à avoir des lignes trop longues. Comptez le nombre de caractères par ligne dans un livre publié par un éditeur professionnel. Vous constaterez qu'il y a environ 66 caractères par ligne. La raison est que la lecture devient difficile dès que l'on dépasse cette limite parce qu'il est alors pénible pour les yeux de se déplacer de la fin de la ligne courante au début de la ligne suivante. *QED.*

SI vous produisez un document PDF à partir d'un fichier RMarkdown vous aurez par défaut des lignes trop longues. Par exemple, le fichier minimaliste ci-après :

```
----- knitr/un.Rmd -----

title: Titre
header-includes:
 \usepackage{lipsum}
output:
 pdf_document

Introduction

\lipsum[1]
```

donnera le résultat suivant :

Titre

### Introduction

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

POUR corriger ce problème il suffit d'indiquer que l'on souhaite pour produire le document PDF utiliser la classe L<sup>A</sup>T<sub>E</sub>X « article » avec du papier au format A4. Le fichier ci-après :

```
----- knitr/trois.Rmd -----

title: Titre
documentclass: article
classoption: a4paper
output:
 pdf_document
header-includes:
 \usepackage{lipsum}

Introduction

\lipsum[1]
```

donnera :

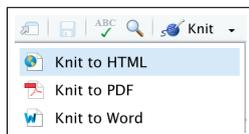
Titre

### Introduction

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

## 3.9 Choix du format de sortie

VOUS avez sans doute remarqué lorsque vous avez choisi « Knit to HTML » que d'autres possibilités vous étaient offertes :



L'IDÉE est qu'à partir d'un même document `RMarkdown` vous puissiez, en théorie, générer des documents aux formats :

**HTML** pour publier votre rapport sur la toile.

**PDF** pour publier votre rapport sous une forme imprimable et non modifiable par ses destinataires.

**Word** pour publier votre rapport sous la forme d'un document modifiable avec un traitement de texte. Pour utiliser un format non propriétaire on préférera dans ce cas de figure l'option `output: odt_document`.

EN pratique, il n'est pas toujours aussi simple que d'avoir un format unique pour générer tous ces types de documents finaux. Par exemple, si vous essayez « Knit to PDF » sur notre document minimaliste `mini.Rmd` vous aurez un message d'erreur :

```
! You can't use '\spacefactor' in math mode.
\@->\spacefactor
 \@m {}
```

CE qui se passe est la chose suivante : dans un fichier destiné à produire un PDF il faut écrire `\LaTeX` pour produire `LATEX`. Mais dans un fichier destiné à produire du HTML écrire `\LaTeX` ne produira rien du tout. Au lieu de cela il faut écrire `LaTeX` comme si c'était une expression mathématique valide, alors qu'elle ne l'est pas. Mais cette écriture va générer l'erreur ci-dessus si on cherche à produire un PDF. Impossible donc à partir d'une même écriture dans le fichier `RMarkdown` de produire à la fois un document HTML et un document PDF à partir du même fichier.

CECI étant dit, comme `R` est un langage de programmation complet, il est tout à fait possible d'écrire ses propres fonctions pour s'adapter automatiquement au format du document produit, dans notre exemple minimaliste cela donnerait par exemple :

```
----- knitr/minib.Rmd -----
output:
 pdf_document: default
 html_document: default

```{r echo=FALSE,results='asis'}
latex <- function() {
  if(knitr::is_latex_output()) {
    sprintf("\\LaTeX")
  } else if(knitr::is_html_output()) {
    sprintf("$\\LaTeX$")
  } else {
    sprintf("LaTeX")
  }
}
...

# Bonjour, ô mon$\delta$e !

Pour vérifier que `r latex()` n'est pas trop obsolète.
```

NOUS n'entrerons pas dans le détail de l'explication de cette solution. Retenez simplement que de générer à la fois des documents aux formats PDF, HTML et Word à partir d'un seul fichier `RMarkdown` n'est pas toujours simple.

3.10 Structuration du document

NOUS avons déjà vu dans la section 3.6 page 24 que # servait à indiquer un titre de niveau 1 et ## un titre de niveau 2. L'option `number_sections: yes` vous permet de numéroter automatiquement les titres. On peut désactiver localement cette option en faisant suivre le titre par {-}. L'exemple suivant reprend la structuration classique d'un article scientifique, notez que le résumé est considéré comme une métadonnée, et donc défini dans le préambule du document.

```
----- knitr/cantatrix.Rmd -----
title: >
  Mise en évidence expérimentale d'une organisation tomatotopique chez la
  soprano (*Cantatrix sopranica* L.)
author: Georges PEREC
output:
  pdf_document:
    number_sections: yes
  html_document:
    number_sections: yes
abstract: >
  *Démonstration expérimentale d'une organisation tomatotopique chez la
  Cantatrice*. L'auteur étudie le lancement de la tomate qui provoquait la
  *réaction yellante* chez la cantatrice et démontre que plusieurs aires de la
  cervelle étaient impliquées dans la réponse, en particulier le trajet légumier,
  les nuclei thalameux et le figure musicien de l'hémisphère nord.
-----

# Introduction{-}

# Matériels et méthodes

## Préparation

## Stimulation

## Enregistrement

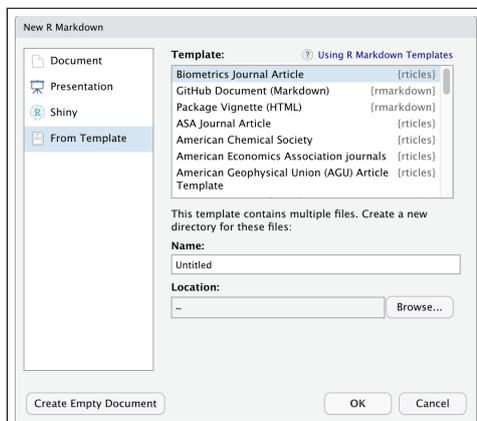
## Histologie

# Résultats

# Discussion

# Références{-}
```

Le paquet `rticles` [3] fournit des modèles prêts à l'emploi pour de nombreuses revues scientifiques, il suffit de le sélectionner au moment de la création du document :



VOICI par exemple à quoi pourrait ressembler un article formaté pour la revue *Biometrics* :

BIOMETRICS 0, 
January 2020

DOI: 0000-0000-0000

Mise en évidence expérimentale d'une organisation tomatotopique chez la soprano (*Cantatrix sopránica* L.)

Georges PEREC

Laboratoire de physiologie, Faculté de médecine Saint-Antoine Paris, France

*email: abc@def

SUMMARY: *Démonstration expérimentale d'une organisation tomatotopique chez la Cantatrice.* L'auteur étudie le lancement de la tomate qui provoquait la *réaction yellante* chez la cantatrice et démontre que plusieurs aires de la cervelle étaient impliquées dans la réponse, en particulier le trajet légumier, les nuclei thalameux et le figure musicien de l'hémisphère nord.

KEY WORDS: tomate soprano cri.

1. Introduction

Les effets frappants du jet de tomates sur les sopranos, observés aux heures ultimes du siècle dernier par Marks et Spencer (1899) qui, les premiers, employèrent le terme de réaction de hurlements (RH), ont été largement décrits dans la littérature. Si de nombreuses études expérimentales (Zeeg & Puss, 1931; Roux & Combaluzier, 1932; Sinon & coll., 1948), anatomopathologique (Hun & Deu, 1960), comparative (Karybb & Szyla, 1973) et prospective (Else & Vire, 1974) ont permis de décrire avec précision ces réponses caractéristiques, les données neuroanatomiques, aussi bien que neurophysiologiques sont, en dépit de leur grand nombre, étonnamment

cortex somesthésique (Pericoloso & Sporghersi, 1973), moteur (Ford, 1930), commissural (Gordon & Bogen, 1974) et associatif (Einstein et coll., 1974) a été évoquée dans le développement progressif de la réponse, bien que, jusqu'à maintenant, les mécanismes afférents et efférents de la RH n'aient jamais été explicités de manière décisive et convaincante.

Unsofort & Tchetera ont observé que *plus on jette de tomates sur les sopranos et plus elles crient*; par ailleurs, des études comparatives, par rapport à la réaction de gasp (Otis & Pifre, 1964), au hoquet (Carpentier & Fialip, 1964), au ronronnement du chat (Remmers & Gautier, 1972), au réflexe HM (Vincent et coll., 1976), à la ventriloquie (Mc Culloch et

LE paquet `bookdown` [103] contient de nombreuses extensions, telle que la gestion des références croisées, qui en font vite un incontournable dès que l'on veut produire un rapport un tant soit peu conséquent.

3.11 Gérer les références bibliographiques

3.11.1 Importance des références bibliographiques

COMMENÇONS par citer Jacques André [5] : « Tout d'abord, voici un extrait des (anciennes) instructions aux auteurs de la revue *TSI – Technique et science informatiques*. »

« Le but des références bibliographiques est double :

1. hommage, honnêteté vis-à-vis des précurseurs, renvoi aux preuves de ce que l'on avance;
2. moyen pour le lecteur de s'informer davantage.

Toute référence doit être exploitable par le lecteur. On citera donc, chaque fois que ce sera possible, un article publié dans une revue plutôt qu'un rapport interne d'université ou d'entreprise; lorsqu'un congrès a fait l'objet d'une publication en volume, c'est celui-ci qu'il convient de citer et non les actes provisoires distribués aux participants.

On s'abstiendra absolument d'inclure des références du type "Dupond, note manuscrite" ou "Dupont, communication personnelle" qui peuvent être avantageusement remplacées, le cas échéant, par une note en bas de page, voire dans un bref paragraphe de remerciements. »

3.11.2 Principe de surjectivité

LE principe de surjectivité auquel doit satisfaire votre document est donné dans la figure 4 page 34. N'essayez pas de maintenir manuellement la surjectivité dans votre document, ce sera pris en charge automatiquement. Ce qui n'est pas automatisable c'est la localisation de l'appel d'une référence. Comparez les assertions suivantes :

1. Pour les échantillons de petite taille il a été proposé [38] d'appliquer une correction, mais elle appliquée maintenant quelle soit la taille de l'échantillon.
2. Pour les échantillons de petite taille il a été proposé d'appliquer une correction [38], mais elle appliquée maintenant quelle soit la taille de l'échantillon.
3. Pour les échantillons de petite taille il a été proposé d'appliquer une correction, mais elle appliquée maintenant quelle soit la taille de l'échantillon [38].

DANS le premier cas Student¹⁰ ne fait qu'une vague proposition, dans le deuxième cas Student donne une correction concrète, dans le troisième cas on fait dire à Student des choses qu'il n'a pas dites. C'est une erreur qu'il est très facile de faire quand on remanie le texte sans penser à repositionner l'appel de références. Aucun outil bibliographique ne peut vous aider ici, c'est votre responsabilité de rédacteur qui est en jeu. Relisez soigneusement le document final.

3.11.3 Le format Bib_TE_X

VOS références bibliographiques doivent être regroupées dans un fichier à part au format Bib_TE_X qui par convention à l'extension *.bib. Pour voir à quoi ressemble ce format il suffit de demander comment citer un paquet , par exemple :

¹⁰Student est le nom de plume de William Seally Gosset, ce qui explique que de façon exceptionnelle il n'y ait pas de prénom dans la référence de son article [38].

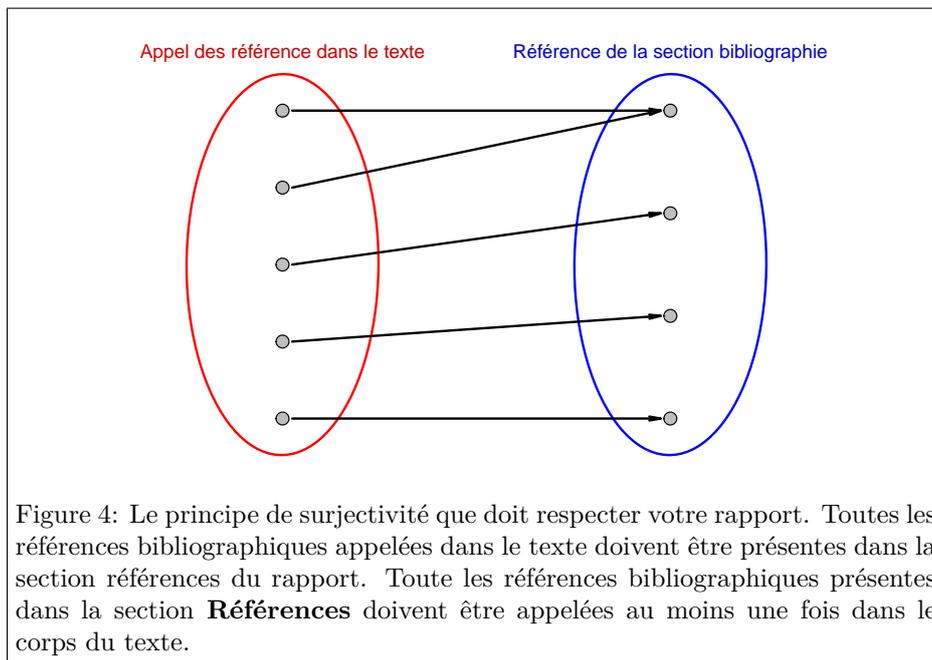


Figure 4: Le principe de surjectivité que doit respecter votre rapport. Toutes les références bibliographiques appelées dans le texte doivent être présentes dans la section références du rapport. Toute les références bibliographiques présentes dans la section **Références** doivent être appelées au moins une fois dans le corps du texte.

```
print(citation("knitr"), bibtex = TRUE)
```

To cite the 'knitr' package in publications use:
 Yihui Xie (2021). knitr: A General-Purpose Package for Dynamic Report
 Generation in R. R package version 1.33.

Une entrée BibTeX pour les utilisateurs LaTeX est

```
@Manual{,
  title = {knitr: A General-Purpose Package for Dynamic Report Generation in R},
  author = {Yihui Xie},
  year = {2021},
  note = {R package version 1.33},
  url = {https://yihui.org/knitr/},
}
```

Yihui Xie (2015) Dynamic Documents with R and knitr. 2nd edition. Chapman
 and Hall/CRC. ISBN 978-1498716963

Une entrée BibTeX pour les utilisateurs LaTeX est

```
@Book{,
  title = {Dynamic Documents with {R} and knitr},
  author = {Yihui Xie},
  publisher = {Chapman and Hall/CRC},
  address = {Boca Raton, Florida},
  year = {2015},
  edition = {2nd},
  note = {ISBN 978-1498716963},
  url = {https://yihui.org/knitr/},
}
```

Yihui Xie (2014) knitr: A Comprehensive Tool for Reproducible Research in
 R. In Victoria Stodden, Friedrich Leisch and Roger D. Peng, editors,
 Implementing Reproducible Computational Research. Chapman and Hall/CRC.
 ISBN 978-1466561595

Une entrée BibTeX pour les utilisateurs LaTeX est

```
@InCollection{,
  booktitle = {Implementing Reproducible Computational Research},
  editor = {Victoria Stodden and Friedrich Leisch and Roger D. Peng},
  title = {knitr: A Comprehensive Tool for Reproducible Research in {R}},
  author = {Yihui Xie},
```

```
publisher = {Chapman and Hall/CRC},
year = {2014},
note = {ISBN 978-1466561595},
url = {http://www.crcpress.com/product/isbn/9781466561595},
}
```

Le mot-clef qui suit immédiatement le caractère @ indique le type du document cité. Nous avons ici un manuel d'utilisation (`Manual`), un livre (`Book`) et un chapitre d'ouvrage (`InCollection`). Pour pouvoir utiliser ces références il manque simplement une clef d'identification juste après l'accolade ouvrante pour pouvoir y faire facilement référence, par exemple :

```
knitr/knitr.bib
@Book{livreKnitr,
  title = {Dynamic Documents with {R} and knitr},
  author = {Yihui Xie},
  publisher = {Chapman and Hall/CRC},
  address = {Boca Raton, Florida},
  year = {2015},
  edition = {2nd},
  note = {ISBN 978-1498716963},
  url = {https://yihui.org/knitr/},
}
```

On indique ensuite dans le préambule avec `bibliography: knitr.bib` que les références sont au format BibTeX dans le fichier `knitr.bib`. On y fait référence avec `[@livreKnitr]` dans le corps du texte. Pour citer plusieurs références on les sépare avec des points-virgule : `[@clef1; @clef2]`.

```
knitr/knitr.Rmd
---
title: "Exemple de citation bibliographique"
output: html_document
bibliography: knitr.bib
---

J'ai utilisé `knitr` [@livreKnitr] pour mettre en forme ce document.

# Références
```

donnera :

Exemple de citation bibliographique

J'ai utilisé `knitr` (Xie 2015) pour mettre en forme ce document.

Références

Xie, Yihui. 2015. *Dynamic Documents with R and Knitr*. 2nd ed. Boca Raton, Florida: Chapman; Hall/CRC.
<https://yihui.org/knitr/>.

3.11.4 Première et seconde main

ATTENTION, ce n'est pas parce que l'auteur A dit que l'auteur B a dit *ceci* que l'auteur B a réellement dit *ceci*. Vous devez autant que faire se peut remonter à la source. Ce n'est pas toujours possible, dans ce cas vous devez faire apparaître clairement que c'est une citation de seconde main. Comparez les deux versions suivantes :

1. Les données utilisées par Student pour ses expériences de simulation [38] sont celles de la table page 219 dans l'article de Macdonell ([62] cité par [38]).

2. Les données utilisées par Student pour ses expériences de simulation [38] sont celles de la table page 216 dans l'article de Macdonell [62].

DANS le premier cas vous indiquez clairement que vous n'avez pas eu accès à la source originale des données, et c'est bien car sinon vous auriez détecté une erreur dans l'article de Student [38] sur la localisation exacte de la table utilisée. Un spécialiste du domaine verra tout de suite si vous essayez de faire passer une citation de seconde main pour une citation de première main. Ne prenez pas de risques : ne citez directement que ce que vous avez pu vérifier par vous-même, donc ce que vous avez lu.

COMME l'écrivait Jacques André [5] : « [o]n a tous vu ces références erronées qui se propagent, par des copier-coller, d'un article à l'autre; sans oublier ces références à des livres ou articles qu'on n'a pas lus mais qu'il est de bon ton de citer. » En tant que vecteur des échanges scientifiques le français étant une langue désuète, ce phénomène est amplifié pour les ouvrages francophones, comme illustré par la figure 5 page 37.

3.11.5 Nomenclatures particulières

UNE nomenclature peut imposer l'usage de caractères particuliers, typiquement l'emploi de *l'italique* pour désigner une espèce. Ainsi, cette référence [10] est correcte alors que celle-ci [9] ne l'est pas. Pour imposer l'emploi de l'italique il faut dans votre fichier *.bib écrire `\textit{Escherichia coli}` pour obtenir *Escherichia coli*. De plus, pour être certain de conserver la casse originelle, il faut encadrer les caractères majuscules avec des accolades, par exemple `{K}-12` pour obtenir K-12. Dans le corps du texte il faut écrire simplement `*Escherichia coli* K-12` pour obtenir *Escherichia coli* K-12. Même si on n'a besoin d'écrire aucune formule mathématique, il est bon d'avoir quelques notions de \LaTeX pour mettre en forme correctement ses références bibliographiques. Imaginez par exemple que vous deviez citer cet article [28] :

Forensic Science International: Genetics 5 (2011) 43–49

Contents lists available at [ScienceDirect](#)



Forensic Science International: Genetics

journal homepage: www.elsevier.com/locate/fsig



One year variability of peak heights, heterozygous balance and inter-locus balance for the DNA positive control of AmpF ℓ STR[®] Identifiler[®] STR kit

A. Debernardi^b, E. Suzanne^b, A. Formant^b, L. Pène^b, A.B. Dufour^a, J.R. Lobry^{a,*}

^a Université de Lyon, F-69000, Lyon ; Université Lyon 1 ; CNRS, UMR 5558, Laboratoire de Biométrie et Biologie Evolutive, F-69622 Villeurbanne, France
^b Institut National de Police Scientifique, Laboratoire de Police Scientifique de Lyon, 31, avenue Franklin Roosevelt, F-69134 Ecully, France

Sans un minimum de connaissance de \LaTeX vous aurez du mal à rendre correctement le « AmpF ℓ STR[®] Identifiler[®] » du titre de cet article. Pour information le « ℓ » est obtenu avec `ℓ` et le sigle du copyright en exposant avec `$\scriptstyle\copyright$` .

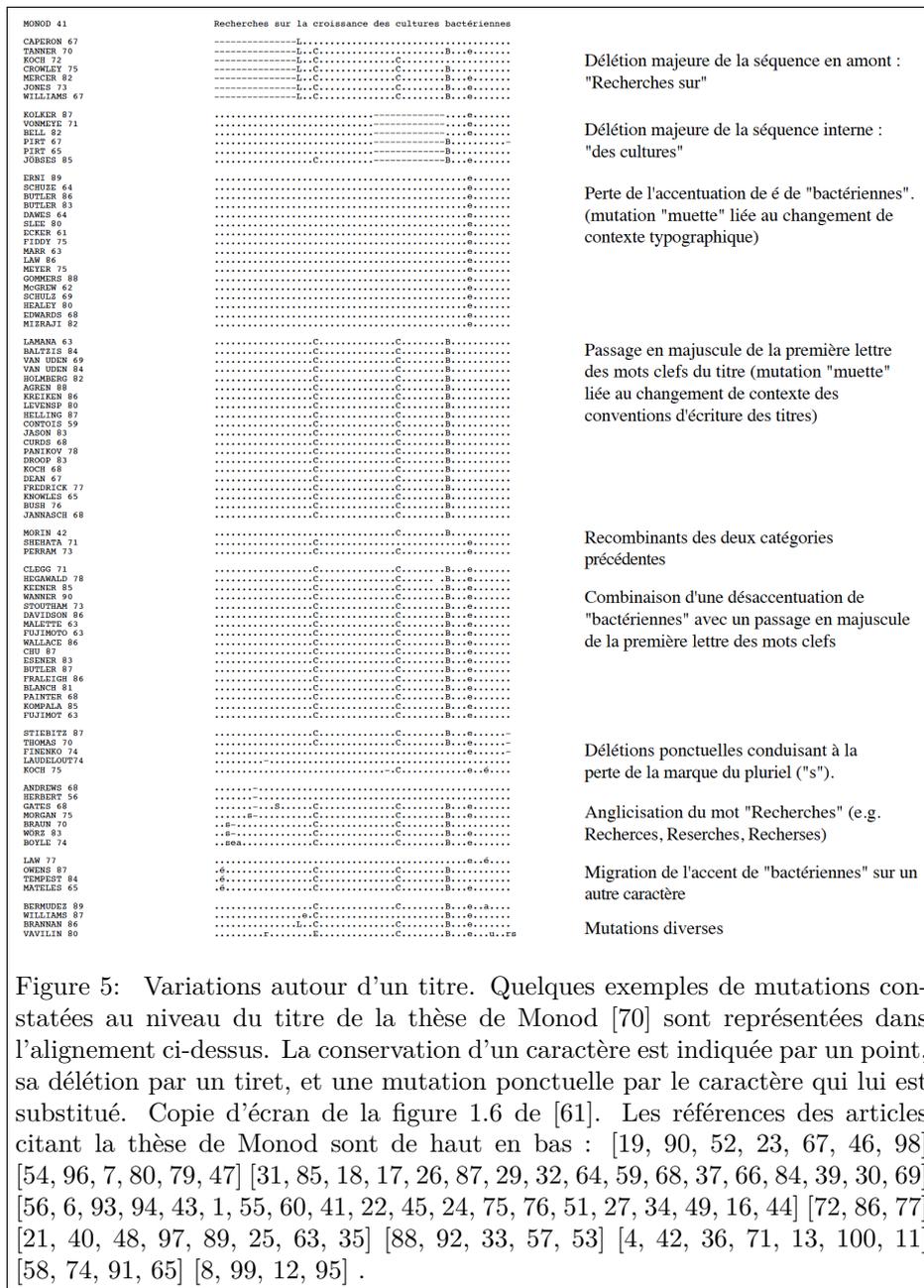


Figure 5: Variations autour d'un titre. Quelques exemples de mutations constatées au niveau du titre de la thèse de Monod [70] sont représentées dans l'alignement ci-dessus. La conservation d'un caractère est indiquée par un point, sa délétion par un tiret, et une mutation ponctuelle par le caractère qui lui est substitué. Copie d'écran de la figure 1.6 de [61]. Les références des articles citant la thèse de Monod sont de haut en bas : [19, 90, 52, 23, 67, 46, 98] [54, 96, 7, 80, 79, 47] [31, 85, 18, 17, 26, 87, 29, 32, 64, 59, 68, 37, 66, 84, 39, 30, 69] [56, 6, 93, 94, 43, 1, 55, 60, 41, 22, 45, 24, 75, 76, 51, 27, 34, 49, 16, 44] [72, 86, 77] [21, 40, 48, 97, 89, 25, 63, 35] [88, 92, 33, 57, 53] [4, 42, 36, 71, 13, 100, 11] [58, 74, 91, 65] [8, 99, 12, 95] .

4 Tricotage

4.1 Insérer du code

C'EST ici que cela va commencer à devenir magique. Considérez l'exemple suivant. C'est encore un fichier **RMarkdown** minimaliste. Nous avons cette fois donné un titre avec `title:` parce qu'il est toujours bon de donner un titre à ses documents. La grande nouveauté c'est que nous avons inséré un fragment de code  encadré par ````${code}```` et ````\n``.

```
knitr/mini2.Rmd
---
title: "Essays de calculer 2 + 2"
output: html_document
---

```${code}```
2 + 2
```\n`
```

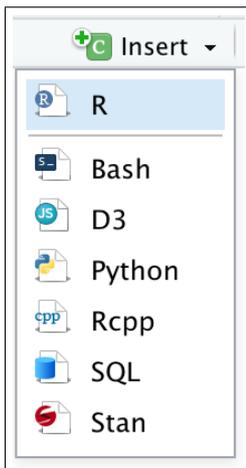
Récupérez le fichier `mini2.Rmd` et sélectionnez « Knit to HTML »:

Essays de calculer 2 + 2

```
2 + 2
```

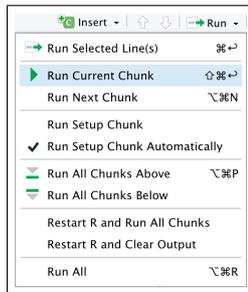
```
## [1] 4
```

Le code  `2 + 2` n'a été entré qu'une seule fois dans le document **RMarkdown**. Dans le document final il est rappelé et le résultat de son exécution est donné. C'est ce qui garanti automatiquement la cohérence entre votre code et les résultats produits. Si vous changez `2 + 2` en `3 + 3`, tout sera mis à jour dans votre rapport. Plus besoin de faire de fastidieuses opérations de copier/coller manuellement. Comme il est assez fastidieux de taper ````${code}```` et ````\n`` pour encadrer le code, **RStudio** vous propose un raccourci clavier (`Ctrl+Alt+I`) pour faire cette opération, ou bien un menu :



UNE fonctionnalité très intéressante de **RStudio** est que quand votre curseur est à l'intérieur d'un fragment de code  vous pouvez utiliser le menu *Run Current Chunk*, ou bien le raccourci clavier correspondant, pour exécuter ce

fragment code immédiatement, sans avoir à recompiler tout votre document. Ceci permet de vérifier de façon très interactive que tout va bien. Si votre fragment de code est dépendant d'autres fragments de code, il faudra les exécuter au préalable pour que cela fonctionne, par exemple avec *Run All*.



POUR ce qui est de la mise en forme du code  on conserve bien entendu un contrôle total du rendu désiré, par exemple :

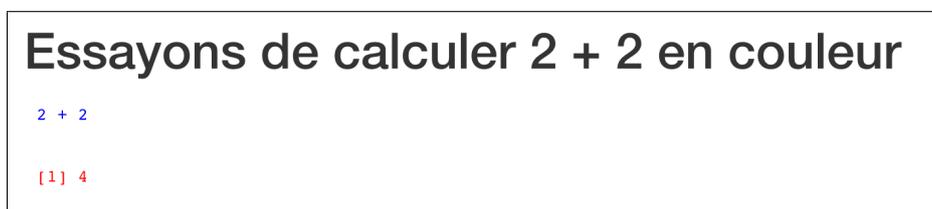
```
knitr/mini3.Rmd
---
title: "Essayons de calculer 2 + 2 en couleur"
output:
  pdf_document: default
  html_document: default
---

```{css, echo=FALSE}
.Rin {
 background-color: white;
 color: blue;
 border: none;
}
.Rout {
 background-color: white;
 color: red;
 border: none;
}
```

```{r setup, echo=FALSE}
knitr::opts_chunk$set(comment = "", class.source = "Rin", class.output = "Rout",
 highlight = FALSE)
```

```{r}
2 + 2
```
```

donnera :



NOUS ne commenterons pas plus cet exemple sauf pour vous faire noter l'option utile `echo=FALSE` qui permet de ne pas faire figurer un fragment de code dans le document final. C'est intéressant si vous souhaitez par exemple alléger le texte reporter en annexe le code  d'une fonction :

```
knitr/mini3b.Rmd
---
title: "La fonction factorielle"
output: html_document
---

```{r setup, echo=FALSE}
knitr::opts_chunk$set(comment = "")
options(scipen = 999) # pour voir tous les chiffres
```

# Corps du rapport

```{r factocode,echo=FALSE}
factorielle <- function(n){
 n <- as.integer(n)
 if(n < 0L) return(NaN)
 if(n == 0L) return(1L)
 return(prod(1:n))
}
```

Pour  $n \in \mathbb{N}$ , la définition par récurrence de la factorielle  $n!$  est :



- $0! = 1$
- Pour tout entier  $n > 0$ ,  $n! = n \times (n - 1)!$



J'ai défini la fonction `factorielle()` (voir annexe pour le code) pour la calculer :

```{r}
factorielle(10)
factorielle(170)
```

# Annexe

```{r eval=FALSE}
<<factocode>>
```
```

donnera :

La fonction factorielle

Corps du rapport

La définition par récurrence de la factorielle est :

1. $0! = 1$
2. Pour tout entier $n > 0$, $n! = (n-1)! \times n$

J'ai défini la fonction `factorielle()` (voir annexe pour le code) pour la calculer :

```
factorielle(10)
```

```
[1] 3628800
```

```
factorielle(100)
```

```
[1] 933262154439441509656467047959538825784009703731840988310128895405822272385704312950661130  
89288327277825849664006524270554535976289719382852181865895959724032
```

Annexe

```
factorielle <- function(n){  
  n <- as.integer(n)  
  if(n == 0){  
    return(1)  
  } else {  
    return(prod(1:n))  
  }  
}
```

NOUS avons donc ici un fragment de code qui s'appelle `factocode` que nous avons utilisé une première fois avec l'option `echo=FALSE` pour qu'il ne figure pas dans le document final mais soit évalué pour que nous puissions utiliser la fonction `factorielle()` qu'il définit. Nous l'avons utilisé une deuxième fois avec `<<factocode>>` avec l'option `eval=FALSE` pour qu'il figure dans l'annexe du rapport mais ne soit pas évalué, puisqu'il l'a déjà été ce serait du gaspillage que l'évaluer une seconde fois pour rien. L'important ici pour assurer la reproductibilité de vos résultats est que le code donné en annexe dans votre rapport pour définir la fonction `factorielle()` soit exactement celui utilisé dans votre rapport. Grâce à `<<factocode>>` c'est garanti automatiquement.

SAUF à ce que votre rapport soit destiné à une audience très technique, il est peu probable que votre lecteur soit intéressé par le détail du code de toutes les fonctions que vous aurez laborieusement écrites. Ce serait une distraction pour lui que d'avoir tous les détails du code dans le corps du texte. Mais pour la *reproductibilité* de vos résultats il est important que tout le code figure en annexe. Nous venons de voir comment procéder dans le cas d'un fragment de code, mais pour de nombreux fragments de code la méthode risque de venir rapidement fastidieuse. Bien heureusement il y a moyen d'automatiser cela et de décider une fois pour toutes de ne publier tous les fragments de code qu'en annexe de votre rapport :

```
----- knitr/factorielle.Rmd -----  
---  
title: "Calcul de la fonction factorielle"  
output: html_document  
---
```

```
```{r setup, echo=FALSE}
knitr::opts_chunk$set(comment = "", echo = FALSE)
```

# Corps du rapport

Pour  $n \in \mathbb{N}$ , la définition par récurrence de la factorielle  $n!$  est :

1.  $0! = 1$ 
2. Pour tout entier  $n > 0$ ,  $n! = n \times (n - 1)!$ 

```{r factocode1}
factorielle1 <- function(n){
 n <- as.integer(n)
 if(n < 0L) return(NaN)
 if(n == 0L) return(1L)
 return(n*factorielle1(n - 1))
}
```

On peut aussi la définir directement :

1.  $0! = 1$ 
2. Pour tout entier  $n > 0$ ,  $n! = \prod_{i=1}^n i$ 

```{r factocode2}
factorielle2 <- function(n){
 n <- as.integer(n)
 if(n < 0L) return(NaN)
 if(n == 0L) return(1L)
 return(prod(1:n))
}
```

Ou passer par la fonction  $\Gamma$  :

1. Pour tout entier  $n \geq 0$ ,  $n! = \Gamma(n+1)$ 

```{r factocode3}
factorielle3 <- function(n){
 n <- as.integer(n)
 if(n < 0L) return(NaN)
 if(n == 0L) return(1L)
 return(gamma(n + 1))
}
```

Voici une comparaison des temps de calcul :

```{r manip}
nrep <- 1000
system.time(replicate(nrep, factorielle1(10)))
system.time(replicate(nrep, factorielle2(10)))
system.time(replicate(nrep, factorielle3(10)))
```

# Annexe

```{r ref.label=knitr::all_labels(), echo=TRUE, eval=FALSE}
```
```

NOUS avons utilisé ici un fragment de code particulier qui s'appelle `setup`. Ce fragment de code sera toujours exécuté en premier, où qu'il soit dans votre document, et sert à initialiser globalement les options. Ici avec la fonction `knitr::opts_chunk$set()` nous avons demandé à ce que par défaut les fragments de code n'apparaissent pas dans le document final (`echo = FALSE`) et nous sommes débarrassés du double croisillon `##` qui préfixe les sorties de code . En annexe nous avons utilisé `ref.label=knitr::all_labels()` pour con-

struire automatiquement un fragment de code concaténant automatiquement tous les fragments de code du document. Si vous souhaitez exclure des annexes certains fragments de code, comme celui du `setup` qui ne présente pas beaucoup d'intérêt pour la reproductibilité de vos résultats, vous pouvez utiliser la fonction `setdiff()` pour éliminer les fragments de code que vous ne voulez pas faire figurer dans votre rapport avec une construction du type :

```
# Annexe

```{r get-labels, echo = FALSE}
labs <- knitr::all_labels()
labs <- setdiff(labs, c("setup", "manip", "get-labels"))
```

```{r ref.label=labs, echo=TRUE, eval=FALSE}
```
```

EN amont de la rédaction du rapport, lorsque l'on est en phase d'analyse ou de développement, certains préfèrent ne travailler que dans un environnement **R** pur, c'est à dire avec un fichier qui ne contient que du code. Il est possible d'intégrer dans votre document des fragments de code provenant d'un fichier externe comme celui-ci :

```
knitr/codexterne.R

## ---- facto-1
factorielle1 <- fonction(n){
  n <- as.integer(n)
  if(n < 0L) return(NaN)
  if(n == 0L) return(1L)
  return(n*factorielle1(n - 1))
}

## ---- facto-2
factorielle2 <- fonction(n){
  n <- as.integer(n)
  if(n < 0L) return(NaN)
  if(n == 0L) return(1L)
  return(prod(1:n))
}

## ---- facto-3
factorielle3 <- fonction(n){
  n <- as.integer(n)
  if(n < 0L) return(NaN)
  if(n == 0L) return(1L)
  return(gamma(n + 1))
}
```

NOTEZ l'utilisation de `## ----` pour délimiter les trois fragments de code `facto-1`, `facto-2` et `facto-3`. Dans le fichier `RMarkdown` on utilise la fonction `knitr::read_chunk()` pour importer ce fichier :

```
knitr/codexterne.Rmd

---
title: "Intégration de code externe"
output: html_document
---

```{r, echo=FALSE, cache=FALSE}
knitr::read_chunk('codexterne.R')
```

Pour  $n$   $\in \mathbb{N}$ , la définition par récurrence de la factorielle  $n!$  est :

1.  $0! = 1$ 
2. Pour tout entier  $n > 0$ ,  $n! = n \times (n - 1)!$ 

```{r facto-1}
```
```

On peut aussi la définir directement :

1. $0! = 1$
2. Pour tout entier $n > 0$, $n! = \prod_{i=1}^n i$

```
```{r facto-2}
...
```
```

Ou passer par la fonction Γ :

1. Pour tout entier $n \geq 0$, $n! = \Gamma(n+1)$

```
```{r facto-3}
...
```
```

Donnera :

Intégration de code externe

Pour $n \in \mathbb{N}$, la définition par récurrence de la factorielle $n!$ est :

1. $0! = 1$
2. Pour tout entier $n > 0$, $n! = n \times (n-1)!$

```
factorielle1 <- function(n){
  n <- as.integer(n)
  if(n < 0L) return(NaN)
  if(n == 0L) return(1L)
  return(n*factorielle1(n - 1))
}
```

On peut aussi la définir directement :

1. $0! = 1$
2. Pour tout entier $n > 0$, $n! = \prod_{i=1}^n i$

```
factorielle2 <- function(n){
  n <- as.integer(n)
  if(n < 0L) return(NaN)
  if(n == 0L) return(1L)
  return(prod(1:n))
}
```

Ou passer par la fonction Γ :

1. Pour tout entier $n \geq 0$, $n! = \Gamma(n+1)$

```
factorielle3 <- function(n){
  n <- as.integer(n)
  if(n < 0L) return(NaN)
  if(n == 0L) return(1L)
  return(gamma(n + 1))
}
```

La seule subtilité ici est qu'il est indispensable de mettre l'option `cache=FALSE` pour le fragment de code qui importe le fichier externe. En effet, il faut savoir que `knitr` est un petit malin qui ne ré-exécutera vos fragments de code que si vous avez fait une modification dans votre fichier `RMarkdown` qui en changerait le résultat. C'est un mécanisme dit de *cache* qui permet de gagner du temps. Cependant, si modifiez votre fichier externe, `knitr` ne le verra pas. Avec `cache=FALSE` on force la lecture du fichier externe à chaque fois pour être certain d'utiliser la dernière version.

4.2 Récupérer le résultat d'un calcul

Il est tout à fait possible, et même souhaitable pour maintenir la cohérence, de récupérer le résultat d'un calcul  et de l'incorporer directement dans votre texte. Si par exemple vous avez des doutes quant à la valeur exacte de $2 + 2$ il vous suffit d'écrire ``r 2 + 2`` dans votre fichier **R**Markdown et vous aurez automatiquement 4 dans le document final. Vous pouvez même utiliser cette expression à l'intérieur d'une formule **LaTeX** et `$2 + 2 = `r 2 + 2`$` vous donnera automatiquement $2 + 2 = 4$. Pour les résultats numériques non entiers,  fournit en général par défaut un nombre de chiffres trop grand pour que cela soit lisible. La fonction `signif()` vous permet d'arrondir facilement à un nombre de chiffres significatifs donné. Voici un exemple complet :

```
----- knitr/mini4.Rmd -----
---
title: "Le volume d'une sphère"
output: html_document
---

```{r setup, echo=FALSE}
knitr::opts_chunk$set(comment = "")
```

Le volume  $V$  d'une sphère de rayon  $r$  est donné par :

$$V(r) = \frac{4}{3}\pi r^3$$


```{r}
Fonction qui calcule le volume d'une sphère :
V <- function(r) 4/3*pi*r^3
Calcul pour une sphère de rayon unité :
V(1)
Avec trois chiffres significatifs :
signif(V(1), 3)
```

Le volume d'une sphère de rayon  $\mathrm{1\text{-}m}$  est donc
de  $\mathrm{r\ signif(V(1), 3)\text{-}m^3}$ .
```

conduisant à :

Le volume d'une sphère

Le volume V d'une sphère de rayon r est donné par :

$$V(r) = \frac{4}{3}\pi r^3$$

```
# Fonction qui calcule le volume d'une sphère :  
v <- function(r) 4/3*pi*r^3  
# Calcul pour une sphère de rayon unité :  
v(1)
```

```
[1] 4.18879
```

```
# Avec trois chiffres significatifs :  
signif(v(1), 3)
```

```
[1] 4.19
```

Le volume d'une sphère de rayon 1 m est donc de 4.19 m³.

REMARQUE : la fonction `toLatex()` du paquet `sfsmisc` vous permet de convertir facilement des valeurs numériques au format scientifique, par exemple

```
 $\pi^{\pi} \approx 1.34 \cdot 10^{18}$  toLatex(pi^(pi*pi), scientific = TRUE, digits = 2)`$!
```

vous donnera :

$\pi^{\pi} \approx 1.34 \cdot 10^{18}$.

4.3 Insérer un tableau

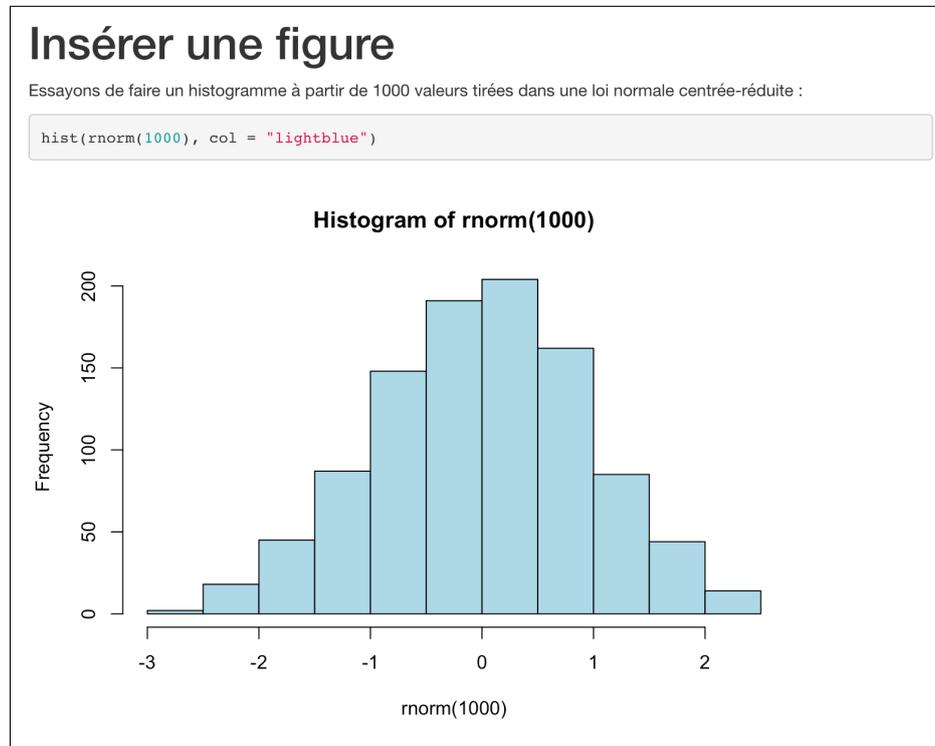
Le fichier `mini5.Rmd` vous donne quelques possibilités pour insérer dans le document final un tableau de données. Le paquet `bookdown` permet de générer des tables auxquelles on peut faire référence dans le texte avec des liens hypertextes.

4.4 Insérer une figure

INSÉRER une figure est particulièrement simple puisque par défaut un fragment de code `R` produisant une figure verra cette dernière automatiquement insérée dans le document final. Par exemple :

```
knitr/mini6.Rmd  
---  
title: "Insérer une figure"  
output: html_document  
---  
  
Essays de faire un histogramme à partir de 1000 valeurs tirées dans une  
loi normale centrée-réduite :  
  
```{r}  
hist(rnorm(1000), col = "lightblue")
```
```

donnera :



LE contrôle de la taille des figures est illustré dans le fichier `mini7.Rmd`. Par défaut `R` utilisera un périphérique graphique de 7 pouces de large par 5 pouces de haut (`fig.dim=c(7,5)`). On peut également modifier la taille du graphique dans le document final avec par exemple `out.width="50%"` pour demander de n'occuper que la moitié de la largeur disponible.

LES figures sont insérées par défaut juste après le code `R` les ayant produites. Il est possible de dissocier les deux. La syntaxe `RMarkdown` pour insérer une image est `![légende](lien){options}`. Voici une façon de procéder :

```

----- knitr/mini6b.Rmd -----
title: "Insérer une figure plus loin"
output: html_document
---
```{r histo,dev='png',fig.show='hide',echo=FALSE}
hist(rnorm(1000), col = "lightblue")
```

Essayons de faire un histogramme à partir de 1000 valeurs tirées dans une
loi normale centrée-réduite :

![Voici mon histogramme](`r knitr::fig_chunk('histo', 'png')`){.class width=50%
    
```

5 Les langages de balisage

5.1 Pas YAML

5.2 \LaTeX

5.3 HTML

5.4 RMarkdown

References

- [1] G.I. Agren. Ideal nutrient productivities and nutrient proportions in plant growth. *Plant Cell and Environment*, 11:613–620, 1988.
- [2] JJ Allaire, Yihui Xie, Jonathan McPherson, Javier Luraschi, Kevin Ushey, Aron Atkins, Hadley Wickham, Joe Cheng, Winston Chang, and Richard Iannone. *rmarkdown: Dynamic Documents for R*, 2020. R package version 2.5.
- [3] JJ Allaire, Yihui Xie, R Foundation, Hadley Wickham, Journal of Statistical Software, Ramnath Vaidyanathan, Association for Computing Machinery, Carl Boettiger, Elsevier, Karl Broman, Kirill Mueller, Bastiaan Quast, Randall Pruim, Ben Marwick, Charlotte Wickham, Oliver Keyes, Miao Yu, Daniel Emaasit, Thierry Onkelinx, Alessandro Gasparini, Marc-Andre Desautels, Dominik Leutnant, MDPI, Taylor and Francis, Oğuzhan Ögreden, Dalton Hance, Daniel Nüst, Petter Uvesten, Elio Campitelli, John Muschelli, Alex Hayes, Zhian N. Kamvar, Noam Ross, Robrecht Cannoodt, Duncan Luguern, David M. Kaplan, Sebastian Kreuzer, Shixiang Wang, Jay Hesselberth, and Christophe Dervieux. *rticles: Article Formats for R Markdown*, 2020. R package version 0.17.
- [4] J.F. Andrews. A mathematical model for the continuous culture of microorganisms utilizing inhibitory substrates. *Biotechnology & Bioengineering*, 10:707–723, 1968.
- [5] J. André. *Petites leçons de typographie*. Editions du jobet, IRISA, 2003-2020.
- [6] B.C. Baltzis and A.G. Fredrickson. Coexistence of two microbial populations competing for a renewable resource in a non-predator prey system. *Bulletin of Mathematical Biology*, 46:155–174, 1984.
- [7] C.R. Bell, M.A. Holder-Franklin, and M. Franklin. Seasonal fluctuations in river bacteria as measured by multivariate analysis of continuous cultures. *Canadian Journal of Microbiology*, 28:959–975, 1982.
- [8] J. Bermudez, D. Lopez, J. Valls, and J. Wagensberg. On the analysis of microbiological processes by Monte Carlo simulation techniques. *CABIOS*, 5:305–312, 1989.
- [9] F.R. Blattner, G. Plunkett III, C.A. Bloch, N.T. Perna, V. Burland, M. Riley, J. Collado-Vides, J.D. Glasner, C.K. Rode, G.F. Mayhew, J. Gregor, N.W. Davis, H.A. Kirkpatrick, M.A. Goeden, D.J. Rose,

- B. Mau, and Y. Shao. The complete genome sequence of *Escherichia coli* K-12. *Science*, 277:1453–1462, 1997.
- [10] F.R. Blattner, G. Plunkett III, C.A. Bloch, N.T. Perna, V. Burland, M. Riley, J. Collado-Vides, J.D. Glasner, C.K. Rode, G.F. Mayhew, J. Gregor, N.W. Davis, H.A. Kirkpatrick, M.A. Goeden, D.J. Rose, B. Mau, and Y. Shao. The complete genome sequence of *Escherichia coli* K-12. *Science*, 277:1453–1462, 1997.
- [11] W.C. Boyle and P.M. Berthouex. Biological wastewater treatment model building fits and misfits. *Biotechnology & Bioengineering*, 16:1139–1159, 1974.
- [12] D.K. Brannan and D.E. Cladwell. Ecology and metabolism of *Thermothrix thiopara*. *Advances in Applied Microbiology*, 31:233–270, 1986.
- [13] H.B. Braun and P.M. Berthouex. Analysis of lag phase BOD curves using the Monod equations. *Water Resources Research*, 6:838–844, 1970.
- [14] R. Bringhurst. *The elements of typographic style*. Hartley & Marks, Vancouver, British Columbia, Canada, 2012.
- [15] J.B. Buckheit and D.L. Donoho. Wavelab and reproducible research. In A. Antoniadis and G. Oppenheim, editors, *Wavelets and Statistics*, volume 103 of *Lecture Notes in Statistics*, pages 55–81. Springer New York, 1995.
- [16] A.W. Bush and A.E. Cook. The effect of time delay and growth rate inhibition in the bacterial treatment of wastewater. *Journal of Theoretical Biology*, 63:385–395, 1976.
- [17] G.J. Butler, S.B. Hsu, and P. Waltman. Coexistence of competing predators in a chemostat. *Journal of Mathematical Biology*, 17:133–151, 1983.
- [18] G.J. Butler and G.S.K. Wolkowicz. Predator-mediated competition in the chemostat. *Journal of Mathematical Biology*, 24:167–191, 1986.
- [19] J. Caperon. Population growth in micro-organisms limited by food supply. *Ecology*, 48:715–722, 1967.
- [20] D. Charif and J.R. Lobry. SeqinR 1.0-2: a contributed package to the R project for statistical computing devoted to biological sequences retrieval and analysis. In H.E. Roman U. Bastolla, M. Porto and M. Vendruscolo, editors, *Structural approaches to sequence evolution: Molecules, networks, populations*, Biological and Medical Physics, Biomedical Engineering, pages 207–232. Springer Verlag, New York, USA, 2007. ISBN 978-3-540-35305-8.
- [21] R.A. Clegg and P.A. Light. Growth yields of *Torulopsis utilis* grown in continuous culture with glycerol or iron as growth limiting nutrient. *Biochemical Journal*, 124:152–154, 1971.
- [22] D.E. Contois. Kinetics of bacterial growth: relationship between population density and specific growth rate of continuous cultures. *Journal of General Microbiology*, 21:40–50, 1959.

- [23] P.H. Crowley. Natural selection and the Michaelis constant. *Journal of Theoretical Biology*, 50:461–475, 1975.
- [24] C.R. Curds and A. Cockburn. Studies on the growth and feeding of *Tetrahymena pyriformis* in axenic and monoxenic culture. *Journal of General Microbiology*, 54:343–358, 1968.
- [25] B.H. Davison and G. Stephanopoulos. Coexistence of *Saccharomyces cerevisiae* and *Escherichia coli* in chemostat under substrate competition and product inhibition. *Biotechnology & Bioengineering*, 28:1742–1752, 1986.
- [26] E.A. Dawes and D.W. Ribbons. Some aspects of the endogenous metabolism of bacteria. *Bacteriological Reviews*, 28:126–149, 1964.
- [27] A.C.R. Dean and P.L. Rogers. The cell size and macromolecular composition of *Aerobacter aerogenes* in various systems of continuous culture. *Biochimica Biophysica Acta*, 148:267–279, 1967.
- [28] A. Debernardi, E. Suzanne, A. Formant, L. Pène, A.B. Dufour, and J.R. Lobry. One year variability of peak heights, heterozygous balance and inter-locus balance for the DNA positive control of AmpF ℓ STR[®] Identifiler[®] STR kit. *Forensic Science International: Genetics*, 5:43–49, 2011.
- [29] R.E. Ecker and W.R. Lockhart. Relationships between initial nutrient concentration and total growth. *Journal of Bacteriology*, 82:80–84, 1961.
- [30] V.H. Edwards and C.R. Wilke. Mathematical representation of batch culture data. *Biotechnology & Bioengineering*, 10:205–232, 1968.
- [31] B. Erni. Glucose transport in *Escherichia coli*. *FEMS Microbiology Reviews*, 63:13–24, 1989.
- [32] C. Fiddy and A.P.S. Trinci. Kinetics and morphology of glucose-limited cultures of moulds grown in a chemostat and on solid media. *Archives of Microbiology*, 103:191–198, 1975.
- [33] Z.Z. Finenko and D.K. Krupatkina-Akinina. Effect of inorganic phosphorus on the growth rate of diatoms. *Marine Biology*, 26:193–201, 1974.
- [34] A.G. Fredrickson. Behavior of mixed cultures of microorganisms. *Annual Review of Microbiology*, 31:63–87, 1977.
- [35] Y. Fujimoto. Kinetics of microbial growth and substrate consumption. *Journal of Theoretical Biology*, 5:171–191, 1963.
- [36] W.E. Gates and J.T. Marlar. Graphical analysis of batch culture data using the Monod expressions. *Journal Water Pollution Control Federation*, 40:R469–R476, 1968.
- [37] P.J.F. Gommers, B.J. van Schie, J.P. van Dijken, and J.G. Kuenen. Biochemical limits to microbial growth yields: an analysis of mixed substrates utilization. *Biotechnology & Bioengineering*, 32:86–94, 1988.

- [38] Student [Gosset, W.S.]. The probable error of a mean. *Biometrika*, 6:1–25, 1908.
- [39] F.P. Healey. Slope of the Monod equation as an indicator of advantage in nutrient competition. *Microbial Ecology*, 5:281–286, 1980.
- [40] E.M. Hegewald and W.A. Knorre. Kinetics of growth and substrate consumption of *Escherichia coli* ML30 on two carbon sources. *Zeitschrift für Allgemeine Mikrobiologie*, 18:415–426, 1978.
- [41] R.B. Helling, C.N. Vargas, and J. Adams. Evolution of *Escherichia coli* during growth in a constant environment. *Genetics*, 116:349–358, 1987.
- [42] D. Herbert, R. Elsworth, and R.C. Telling. The continuous culture of bacteria; a theoretical and experimental study. *Journal of General Microbiology*, 14:601–622, 1956.
- [43] A. Holmberg. On the practical identifiability of microbial growth models incorporating Michaelis-Menten type nonlinearities. *Mathematical Biosciences*, 62:23–43, 1982.
- [44] H.W. Jannasch. Bacterial growth at low population densities. *Nature*, 196:496–497, 1962.
- [45] A.C. Jason. A deterministic model for monophasic growth of batch cultures of bacteria. *Antonie van Leeuwenhoek*, 49:513–536, 1983.
- [46] G.L. Jones. Role of protozoa in waste purification systems. *Nature*, 243:546–547, 1973.
- [47] I.M.L. Jöbses, G.T.C. Egberts, A. van Baalen, and J.A. Roels. Mathematical modeling of growth and substrate conversion of *Zymomonas mobilis* at 30 °c and 35 °c. *Biotechnology & Bioengineering*, 27:984–995, 1985.
- [48] J.P. Keener. Oscillatory coexistence in a food chain model with competing predators. *Journal of Mathematical Biology*, 22:123–135, 1985.
- [49] G. Knowles, A.L. Downing, and M.J. Barrett. Determination of kinetic constants for nitrifying bacteria in mixed culture, with the aid of an electronic computer. *Journal of General Microbiology*, 38:263–278, 1965.
- [50] D.E. Knuth. Literate programming. *The Computer Journal*, 27:97–111, 1984.
- [51] A.L. Koch. Theory of the angular dependance of light scattered by bacteria and similar sized biological objects. *Journal of Theoretical Biology*, 18:133–156, 1968.
- [52] A.L. Koch. Deviations from hyperbolic dependency of transport processes. *Journal of Theoretical Biology*, 36:23–40, 1972.
- [53] A.L. Koch. Lag in adaptation to lactose as a probe to the timing of permease incorporation into the cell membrane. *Journal of Bacteriology*, 124:435–444, 1975.

- [54] Y. Kolker. A piecewise-linear growth model: comparison with competing forms in batch culture. *Journal of Mathematical Biology*, 25:543–551, 1987.
- [55] E. Kreikenbohm, R. Bohl. A mathematical model of syntrophic cocultures in the chemostat. *FEMS Microbiology Ecology*, 38:131–140, 1986.
- [56] C. Lamanna. Studies on endogenous metabolism in bacteriology. *Annals of the New York Academy of Sciences*, 102:517–520, 1963.
- [57] H. Laudelout, R. Lambert, J.L. Fripiat, and L.M. Pham. Effet de la température sur la vitesse d'oxydation de l'ammonium en nitrate par des cultures mixtes de nitrifiants. *Annales de Microbiologie de l'Institut Pasteur*, 125B:75–84, 1974.
- [58] A.T. Law and D.K. Button. Multiple-carbon-source-limited growth kinetics of a marine coryneform organism. *Journal of Bacteriology*, 129:115–123, 1977.
- [59] A.T. Law and D.K. Button. Modulation of affinity of marine pseudomonad for toluene and benzene by hydrocarbon exposure. *Applied and Environmental Microbiology*, 51:469–476, 1986.
- [60] O. Levenspiel. The Monod equation: a revisit and a generalization to product inhibition situations. *Biotechnology & Bioengineering*, 22:1671–1687, 1980.
- [61] J.R. Lobry. *Ré-évaluation de modèle de croissance de Monod. Effet des antibiotiques sur l'énergie de maintenance*. PhD thesis, University Claude Bernard, Lyon I, 1991.
- [62] W.R. Macdonell. On criminal anthropometry and the identification of criminals. *Biometrika*, 1:177–227, 1902.
- [63] M.F. Mallette. Validity of the concept of energy of maintenance. *Annals of the New York Academy of Sciences*, 102:521–535, 1963.
- [64] A.G. Marr, E.H. Nilson, and D.J. Clark. The maintenance requirement of *Escherichia coli*. *Annals of the New York Academy of Sciences*, 102:536–548, 1963.
- [65] R.I. Mateles and E. Battat. Continuous culture used for media optimisation. *Applied Microbiology*, 28:901–905, 1974.
- [66] S.B. McGrew and M.F. Mallette. Energy of maintenance in *Escherichia coli*. *Journal of Bacteriology*, 83:844–850, 1962.
- [67] L.P. Mercer, S.M. Behr, and J. Carter. A mathematical model for nutrition in toxicology. *Nutrition Reports International*, 26:115–120, 1982.
- [68] R. Meyer. Zur Theorie des Keimwachstums als Funktion der Zeit und der Konzentration. *Zentralblatt für Bakteriologie Mikrobiologie und Hygiene Abteilung Series A - Medical Microbiology Infectious Diseases Virology Parasitology*, 232:512–520, 1975.

- [69] E. Mizraji and J. Lin. Migration of enzymes on linear polymers. *Journal of Theoretical Biology*, 96:561–569, 1982.
- [70] J. Monod. *Recherches sur la croissance des cultures bactériennes*. PhD thesis, Paris, 1941.
- [71] P.H. Morgan, L.P. Mercer, and N.W. Flodin. General model for nutritional responses of higher organisms. *Proceedings of the National Academy of Sciences of the United States of America*, 72:4327–4331, 1975.
- [72] F. Morin and J. Monod. Sur l’expression analytique de la croissance des populations bactériennes. *La Revue Scientifique*, 1942:227–229, 1942.
- [73] Denis Mukhin, David A. James, and Jake Luciani. *ROracle: OCI Based Oracle Database Interface for R*, 2016. R package version 1.3-1.
- [74] J.D. Owens and J.D. Legan. Determination of the Monod substrate saturation constant for microbial growth. *FEMS Microbiology Reviews*, 46:419–432, 1987.
- [75] N. Panikov and S.J. Pirt. The effects of cooperativity and growth yield variation on the kinetics of nitrogen or phosphate limited growth of chlorella in chemostat culture. *Journal of General Microbiology*, 108:295–303, 1978.
- [76] N. Panikov and S.J. Pirt. 25 years of algal growth kinetics. a personal view. *Botanica Marina*, 26:99–112, 1983.
- [77] J.W. Perram. Relaxation times in bacteriological culture and the approach to steady state. *Journal of Theoretical Biology*, 38:571–578, 1973.
- [78] E. Peña. Calculating line length: an arithmetic approach. *Visible Language*, 50:112–125, 2016.
- [79] S.J. Pirt. The maintenance energy of bacteria in growing cultures. *Proceedings of the Royal Society of London*, 163B:224–231, 1965.
- [80] S.J. Pirt. A kinetic study of the mode of growth of surface colonies of bacteria and fungi. *Journal of General Microbiology*, 47:181–197, 1967.
- [81] N. Ramsey. Literate programming simplified. *IEEE Software*, 11:97–105, 1994.
- [82] L. Rello and R. Baeza-Yates. Good fonts for dyslexia. In *ASSETS’13*, pages 14–21, Bellevue, WA, USA, 2013. ACM Press.
- [83] L. Rello and R. Baeza-Yates. The effect of font type on screen readability by people with dyslexia. *ACM Transactions on Accessible Computing*, 8:15–33, 2016.
- [84] J.S. Schultz and P. Gerhardt. Dialysis culture of microorganisms: Design, theory, and results. *Bacteriological Reviews*, 33:1–47, 1969.
- [85] K.L. Schulze and R.S. Lipe. Relationship between substrate concentration, growth rate and respiration rate of *Escherichia coli* in continuous culture. *Archives of Microbiology*, 48:1–20, 1964.

- [86] T.E. Shehata and A.G. Marr. Effect of nutrient concentration on the growth of *Escherichia coli*. *Journal of Bacteriology*, 107:210–216, 1971.
- [87] A.M. Slee and J.M. Tanzer. Effect of growth conditions on sucrose phosphotransferase activity of streptococcus mutans. *Infection and Immunity*, 27:922–927, 1980.
- [88] O. Stiebitz, K.H. Wolf, and W. Kloeden. Auswirkungen der Methode der Linearisierung des μ -Cs-Verlaufs auf die Parameteranpassung bei dem Monodschen Modell. *Acta Biotechnologica*, 7:167–172, 1987.
- [89] A.H. Stouthamer and C. Bettenhausen. Utilization of energy for growth and maintenance in continuous and batch culture of microorganisms. *Biochimica Biophysica Acta*, 301:53–70, 1973.
- [90] R.D. Tanner. An enzyme kinetic model for describing fermentation processes. *Biotechnology & Bioengineering*, 12:831–843, 1970.
- [91] D.W. Tempest and O.M. Neijssel. The status of y_{ATP} and maintenance energy as biologically interpretable phenomena. *Annual Review of Microbiology*, 38:459–486, 1984.
- [92] W.H. Thomas. Effect of ammonium and nitrate concentration on chlorophyll increases in natural tropical pacific phytoplankton populations. *Limnology Oceanography*, 15:386–394, 1970.
- [93] N. van Uden. Kinetics of nutrient-limited growth. *Annual Review of Microbiology*, 23:473–486, 1969.
- [94] N. van Uden. Temperature profiles of yeast. *Advances in Microbial Physiology*, 25:195–251, 1984.
- [95] V.A. Vavilin and V.B. Vasiliev. Nonlinear models of biological treatment processes. *Ecological Modelling*, 9:57–71, 1980.
- [96] K. Von Meyenburg. Transport-limited growth rates in a mutant of *Escherichia coli*. *Journal of Bacteriology*, 107:878–888, 1971.
- [97] U. Wanner and T. Egli. Dynamics of microbial growth and cell composition in batch culture. *FEMS Microbiology Reviews*, 75:19–44, 1990.
- [98] F.M. Williams. A model of cell growth dynamics. *Journal of Theoretical Biology*, 15:190–207, 1967.
- [99] T.G. Williams and D.H. Turpin. Photosynthetic kinetics determine the outcome of competition for dissolved inorganic carbon by freshwater microalgae: implication for acidified lakes. *Ecologia*, 73:307–311, 1987.
- [100] A. Wörz-Busekros and P. Lange. Dynamics of a continuous culture with catalysis of the growth-limiting substrate by an enzyme of the cells. *Bulletin of Mathematical Biology*, 45:981–990, 1983.
- [101] Yihui Xie. knitr: A comprehensive tool for reproducible research in R. In Victoria Stodden, Friedrich Leisch, and Roger D. Peng, editors, *Implementing Reproducible Computational Research*. Chapman and Hall/CRC, 2014. ISBN 978-1466561595.

- [102] Yihui Xie. *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition, 2015. ISBN 978-1498716963.
- [103] Yihui Xie. *bookdown: Authoring Books and Technical Documents with R Markdown*. Chapman and Hall/CRC, Boca Raton, Florida, 2016. ISBN 978-1138700109.
- [104] Yihui Xie. Tinytex: A lightweight, cross-platform, and easy-to-maintain latex distribution based on tex live. *TUGboat*, (1):30–32, 2019.
- [105] Yihui Xie. *knitr: A General-Purpose Package for Dynamic Report Generation in R*, 2020. R package version 1.30.
- [106] Yihui Xie. *tinytex: Helper Functions to Install and Maintain TeX Live, and Compile LaTeX Documents*, 2020. R package version 0.27.
- [107] Yihui Xie, J.J. Allaire, and Garrett Grolemond. *R Markdown: The Definitive Guide*. Chapman and Hall/CRC, Boca Raton, Florida, 2018. ISBN 9781138359338.
- [108] Yihui Xie, Christophe Dervieux, and Emily Riederer. *R Markdown Cookbook*. Chapman and Hall/CRC, Boca Raton, Florida, 2020. ISBN 9780367563837.
- [109] Achim Zeileis, the R community. Contributions (fortunes and/or code) by Torsten Hothorn, Peter Dalgaard, Uwe Ligges, Kevin Wright, Martin Maechler, Kjetil Brinchmann Halvorsen, Kurt Hornik, Duncan Murdoch, Andy Bunn, Ray Brownrigg, Roger Bivand, Spencer Graves, Jim Lemon, Christian Kleiber, David L. Reiner, Berton Gunter, Roger Koenker, Charles Berry, Marc Schwartz, Michael Dewey, Ben Bolker, Peter Dunn, Sarah Goslee, Simon Blomberg, Bill Venables, Roland Rau, Thomas Petzoldt, Rolf Turner, Mark Leeds, Emmanuel Charpentier, Chris Evans, Paolo Sonego, Peter Ehlers, Detlef Steuer, Tal Galili, Greg Snow, Brian D. Ripley, Michael Sumner, David Winsemius, Liviu Andronic, Brian Diggs, Matthieu Stigler, Michael Friendly, Dirk Eddelbuettel, Richard M. Heiberger, Patrick Burns, Dieter Menne, Andrie de Vries, Barry Rowlingson, Renaud Lancelot, R. Michael Weylandt, Jon Olav Skoien, Francois Morneau, Antony Unwin, Joshua Wiley, Terry Therneau, Bryan Hanson, Henrik Singmann, Eduard Szoecs, Gregor Passolt, and John C. Nash. *fortunes: R Fortunes*, 2016. R package version 1.5-4.