

<https://github.com/DanChaltiel/autoimport>

Autoimport

Pour gérer l'enfer des @importFrom



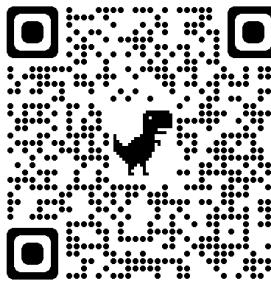
Dan Chaltiel
Biostatisticien
Institut Gustave Roussy



Oncostat



Centre de recherche en épidémiologie et santé des populations



autoimport: contexte

- Crédit à la communauté R pour l'outil autoimport
- Création de packages 📦 avec dépendances (imports)
 - Documenté dans [R Packages \(2^e\)](#)
 - Basé sur le package {roxygen2}
- Solution 1: importer tout un package

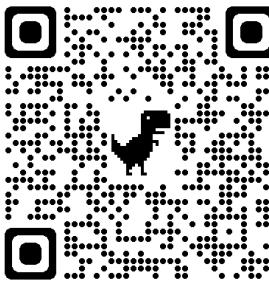
```
#' @import rlang
```

After calling `devtools::document()`, this roxygen tag causes this directive to appear in the NAMESPACE file:

```
import(rlang)
```

This is the least recommended solution because it can make your code harder to read (you can't tell where a function is coming from), and if you `@import` many packages, it increases the chance of function name conflicts. Save this for very special situations.





autoimport: contexte

- Solution 2 :

- Utiliser les préfixes partout dans le code: `pkg::fun()`
- Méthode recommandée par défaut
- MAIS rend le code plus long et moins lisible



- Solution 3 :

- Importer seulement les fonctions : `@importFrom pkg fun1 fun2`
- Sub-question : où mettre cette mention ?
 - Centralisée dans la documentation « package-level »
 - Au plus près de l'appel de fonction





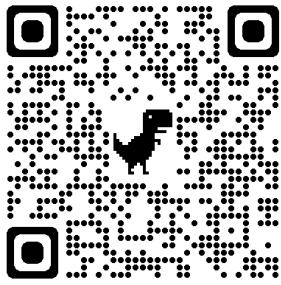
autoimport: concept

- Centraliser rend les fonctions moins portables 
- Mettre @importFrom au dessus de chaque fonction qui l'appelle

```
1 #' @importFrom dplyr across any_of everything
2 #' @importFrom ...
3 #' @importFrom tidyverse unite
4 crosstable = function( ... )
5
6
7 #' @importFrom glue glue
8 #' @importFrom ...
9 #' @importFrom purrr map
10 as_flextable = function( ... )
```

- Un ENFER à maintenir à la main !





autoimport: step 1

- 1^{ère} étape: *reading*
- Lit toutes les fonctions de tous les fichiers sous le chemin "R/"
- Utilise le parser interne de R (parse() & getSource())

```
> autoimport::autoimport()

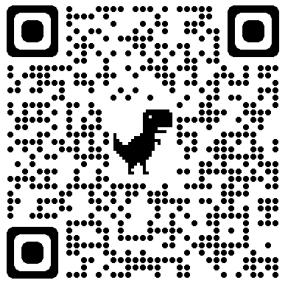
-- Reading --
i Found 2 functions in file R/burgled.R (77 lines)
i Found 8 functions in file R/compact.R (170 lines)
i Found 1 function in file R/cross_by.R (108 lines)
i Found 15 functions in file R/cross_categorical.R (376 lines)
i Found 4 functions in file R/cross_numeric.R (156 lines)
i Found 3 functions in file R/cross_survival.R (169 lines)
i Found 1 function in file R/crosstable-package.R (13 lines)
i Found 3 functions in file R/crosstable.R (468 lines)
i Found 2 functions in file R/dataset.R (76 lines)
i Found 15 functions in file R/effect.R (413 lines)
i Found 9 functions in file R/flextable.R (396 lines)
i Found 13 functions in file R/funs.R (295 lines)
i Found 3 functions in file R/gt.R (122 lines)
i Found 14 functions in file R/labels.R (402 lines)
i Found 5 functions in file R/methods.R (36 lines)
i Found 21 functions in file R/officer.R (897 lines)
i Found 2 functions in file R/openxlsx.R (164 lines)
i Found 3 functions in file R/options.R (186 lines)
i Found 3 functions in file R/pivot.R (158 lines)
i Found 8 functions in file R/reexport.R (32 lines)
i Found 12 functions in file R/test.R (444 lines)
i Found 32 functions in file R/utils.R (688 lines)
v Found a total of 179 internal functions in 22 files (5846 lines).
```



autoimport: step 2

- 2^{ème} étape: *parsing*
- Parse toutes les fonctions trouvées pour détecter l'appel de fonctions externes
- Système de cache pour accélérer les appels futurs
- Utilise le parser interne de R (`parse()` & `getParseData()`)

```
-- Parsing --
> File R/burgled.R
i  Parsing code block `unnamed_1`
i  Parsing code block `unnamed_2`
! Updating cache
i Found 33 functions to import in 2 functions or code chunks.
> File R/compact.R
i  Parsing function `ct_compact()`
i  Parsing function `ct_compact.data.frame()`
i  Parsing function `ct_compact.crosstable()`
i  Parsing function `ct_compact.default()`
i  Parsing function `compact()`
i  Parsing function `compact.data.frame()`
i  Parsing function `compact.crosstable()`
i  Parsing function `compact.default()`
! Updating cache
i Found 49 functions to import in 8 functions or code chunks.
i Found 227 functions to import in 32 functions or code chunks.
v Found a total of 1561 potential functions to import
```



autoimport: step 3

- 3ème étape: *attributing*
- Pour les fonctions équivoques, on demande à l'utilisateur
- Système de fichier pour retenir les choix (IMPORTLIST)

```
-- Attributing --
i Automatically attributing 2 functions imports from inst/IMPORTLIST

There are 3 functions that can be imported from several packages. What do you want to do?

1: Choose the package for each
2: Choose for me please
3: Abort mission

Selection: 1
`align()` can be found in several packages.
From which one do you want to import it:

1: pillar (0 function imported)
2: xtable (0 function imported)

`filter()` can be found in several packages.
From which one do you want to import it:

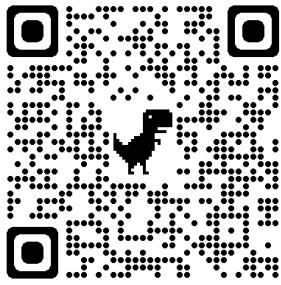
1: stats (6 functions imported)
2: dplyr (3 functions imported)

Selection: 2

Do you want to save your choices about these 3 functions in `inst/IMPORTLIST`?

1: Yes
2: No

Selection: 1
i 3 lines added to inst/IMPORTLIST
```

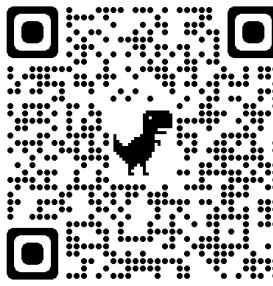


autoimport: step 4

- 4^{ème} étape: *writing*
- Réécrit les imports dans des fichiers temporaires
- Réorganise les imports par ordre alphabétique

```
-- Writing --
i 2 inserts in R/burgled.R
> Nothing done in R/burgled.R (all is already OK)
i 16 inserts in R/compact.R
v Added 8 and removed 3 lines from R/compact.R.
i 7 inserts in R/cross_by.R
v Added 2 and removed 1 line from R/cross_by.R.
i 29 inserts in R/cross_categorical.R
v Added 8 and removed 7 lines from R/cross_categorical.R.
i 18 inserts in R/cross_numeric.R
v Added 5 and removed 4 lines from R/cross_numeric.R.
i 17 inserts in R/cross_survival.R
v Added 4 and removed 3 lines from R/cross_survival.R.
v Ignoring R/crosstable-package.R. Use `ignore_package=FALSE` to override.
i 13 inserts in R/crosstable.R
v Added 5 and removed 3 lines from R/crosstable.R.
i 0 inserts in R/dataset.R
> Nothing done in R/dataset.R (all is already OK)
i 28 inserts in R/effect.R
v Added 4 and removed 3 lines from R/effect.R.
i 21 inserts in R/flextable.R
v Added 6 and removed 4 lines from R/flextable.R.
```

```
-- Finished --
v To view the diff and choose whether or not accepting the changes, run:
i autoimport::import_review()
```



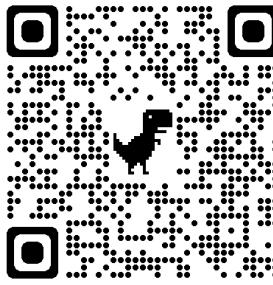
autoimport: review

- `autoimport::import_review()`
- En cliquant sur « Accept », on vient remplacer les lignes générées automatiquement
- Utilise l'appli shiny du package `diffviewer`, inspiré de `testthat`

```
compact.R [CHANGED]
@@ -24,14 +24,17 @@
24 24 #' @rdname ct_compact
25 25 #
26 26 #' @author Dan Chaltiel
27 27 #' @export
28 28 #' @importFrom checkmate assert_scalar
29 - #' @importFrom dplyr across any_of everything lag mutate row_number select sym
30 - #' @importFrom flextable align bold border
+ #' @importFrom crayon bold
+ #' @importFrom dplyr across lag mutate row_number select
31 + #' @importFrom error align
32 #' @importFrom officer fp_border
33 + #' @importFrom rlang sym
34 #' @importFrom tidyverse replace_na
35 + #' @importFrom tidyselect any_of everything
36 #'
```



autoimport: conclusion



- Package encore très expérimental
 - Quelques bugs en cours de création
 - Quelques bugs en cours d'incorporation
- Bien utile en pratique
- Suggestions et PR bienvenues 😊

Merci de votre attention

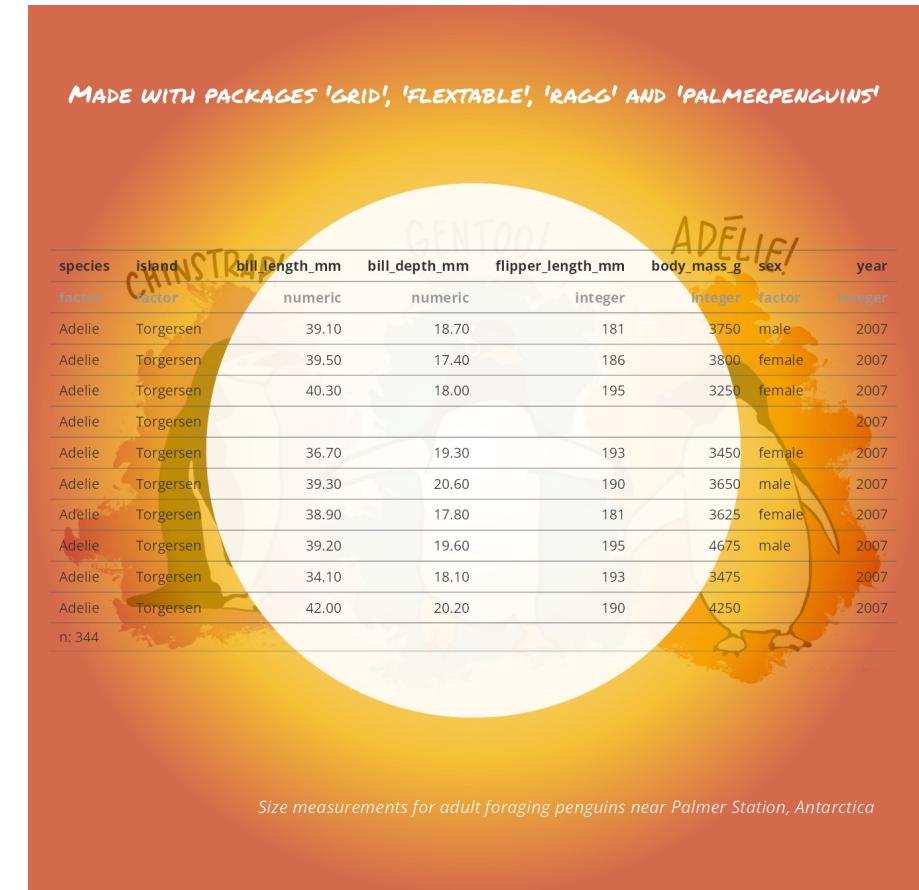
Réaliser ses tableaux avec flextable

Les avancées liées au Rconsortium

'flextable' (2017 - Gohel et Skintzos) fournit une grammaire pour créer et personnaliser des tableaux.

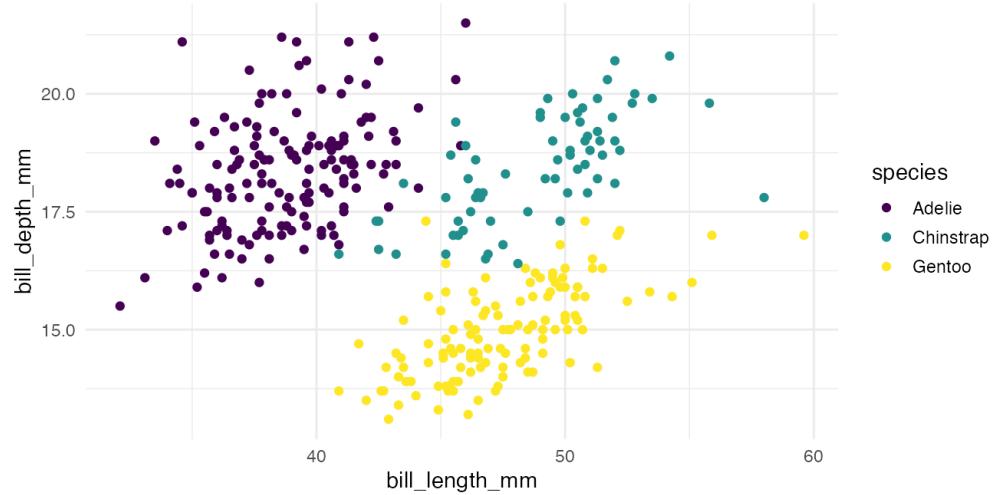
Les formats suivants sont pris en charge : 'Word' (.docx), 'PowerPoint' (.pptx), 'RTF', 'PDF', 'HTML' et R 'Grid Graphics'. La syntaxe est la même pour l'utilisateur quel que soit le type de sortie à produire.

Le package offre également un ensemble de fonctions de haut niveau qui permettent, par exemple, la présentation sous forme de tableaux de modèles statistiques et la création de tableaux croisés complexes.



Les dernières avancées - grid graphics

Été 2022 : sortie graphique complètement intégrée avec grid (et donc ggplot2, patchwork et cowplot)



species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex	year
factor	factor	numeric	numeric	integer	integer	factor	integer
Adelie	Torgersen	39.10	18.70	181	3750	male	2007
Adelie	Torgersen	39.50	17.40	186	3800	female	2007
Adelie	Torgersen	40.30	18.00	195	3250	female	2007
Adelie	Torgersen						2007
Adelie	Torgersen	36.70	19.30	193	3450	female	2007
Adelie	Torgersen	39.30	20.60	190	3650	male	2007
Adelie	Torgersen	38.90	17.80	181	3625	female	2007
Adelie	Torgersen	39.20	19.60	195	4675	male	2007
Adelie	Torgersen	34.10	18.10	193	3475		2007
Adelie	Torgersen	42.00	20.20	190	4250		2007

n: 344

Les dernières avancées – RTF

flextable and RTF

Recours au 49-3

Historique des recours au 49-3 pendant la V république ⁽¹⁾⁽²⁾			
Premier ministre	Recours au 49-3 ⁽³⁾	Textes concernés ⁽⁴⁾	Motions de censure ⁽³⁾
Élisabeth Borne	● 2022-2023 (313 jours)	11	3
Jean Castex	● 2020-2022 (682 jours)		14
Édouard Philippe	● 2017-2020 (1145 jours)	1	1
Bernard Cazeneuve	● 2016-2017 (155 jours)		2
Manuel Valls	● 2014-2016 (981 jours)	6	2
Jean-Marc Ayrault	● 2012-2014 (685 jours)		3
François Fillon	● 2007-2012 (1820 jours)		
Dominique de Villepin	● 2005-2007 (712 jours)	1	1
Jean-Pierre Raffarin	● 2002-2005 (1121 jours)	2	2
Lionel Jospin	● 1997-2002 (1799 jours)		2
Alain Juppé	● 1995-1997 (747 jours)	2	2
Édouard Balladur	● 1993-1995 (773 jours)	1	1
Pierre Bérégovoy	● 1992-1993 (361 jours)	3	1
Edith Cresson	● 1991-1992 (323 jours)	8	2
Michel Rocard	● 1988-1991 (1100 jours)	28	5

page 1

flextable and RTF

Historique des recours au 49-3 pendant la V république⁽¹⁾⁽²⁾

Premier ministre	Recours au 49-3 ⁽³⁾	Textes concernés ⁽⁴⁾	Motions de censure ⁽³⁾
Jacques Chirac	● 1986-1988 (782 jours)	8	7
Laurent Fabius	● 1984-1986 (611 jours)	4	2
Pierre Mauroy	● 1981-1984 (1153 jours)	7	5
Raymond Barre	● 1976-1981 (1722 jours)	8	5
Jacques Chirac	● 1974-1976 (821 jours)		13
Pierre Messmer	● 1972-1974 (691 jours)		
Jacques Chaban-Delmas	● 1969-1972 (1111 jours)		
Maurice Couve de Murville	● 1968-1969 (345 jours)		
Georges Pompidou	● 1962-1968 (2279 jours)	6	2
Michel Debré	● 1959-1962 (1192 jours)	4	4

⁽¹⁾ source : <https://www.assemblee-nationale.fr>

⁽²⁾ source : <https://www.politiquemania.com/longevite-premier-ministre.html>

⁽³⁾Fréquence ⁽⁴⁾Nombre de cas;

page 2

Q1 2023 : sortie RTF implémentée (pour la pharma) – proche des sorties Word et compatibilité complète (images + textes)

Les dernières avancées - refonte de proc_freq

```
set_flextable_defaults(
  font.family = "Open Sans",
  font.size = 9)
```

```
with(ggplot2::diamonds,
  table(clarity, cut)) |>
  as_flextable()
```

OU

```
proc_freq(
  x = ggplot2::diamonds,
  row = "clarity",
  col = "cut")
```

Var1		Var2					Total
		Fair	Good	Very Good	Premium	Ideal	
I1	Count	210 (0.4%)	96 (0.2%)	84 (0.2%)	205 (0.4%)	146 (0.3%)	741 (1.4%)
	Mar. pct ⁽¹⁾	13.0% ; 28.3%	2.0% ; 13.0%	0.7% ; 11.3%	1.5% ; 27.7%	0.7% ; 19.7%	
SI2	Count	466 (0.9%)	1,081 (2.0%)	2,100 (3.9%)	2,949 (5.5%)	2,598 (4.8%)	9,194 (17.0%)
	Mar. pct	28.9% ; 5.1%	22.0% ; 11.8%	17.4% ; 22.8%	21.4% ; 32.1%	12.1% ; 28.3%	
SI1	Count	408 (0.8%)	1,560 (2.9%)	3,240 (6.0%)	3,575 (6.6%)	4,282 (7.9%)	13,065 (24.2%)
	Mar. pct	25.3% ; 3.1%	31.8% ; 11.9%	26.8% ; 24.8%	25.9% ; 27.4%	19.9% ; 32.8%	
VS2	Count	261 (0.5%)	978 (1.8%)	2,591 (4.8%)	3,357 (6.2%)	5,071 (9.4%)	12,258 (22.7%)
	Mar. pct	16.2% ; 2.1%	19.9% ; 8.0%	21.4% ; 21.1%	24.3% ; 27.4%	23.5% ; 41.4%	
VS1	Count	170 (0.3%)	648 (1.2%)	1,775 (3.3%)	1,989 (3.7%)	3,589 (6.7%)	8,171 (15.1%)
	Mar. pct	10.6% ; 2.1%	13.2% ; 7.9%	14.7% ; 21.7%	14.4% ; 24.3%	16.7% ; 43.9%	
VVS2	Count	69 (0.1%)	286 (0.5%)	1,235 (2.3%)	870 (1.6%)	2,606 (4.8%)	5,066 (9.4%)
	Mar. pct	4.3% ; 1.4%	5.8% ; 5.6%	10.2% ; 24.4%	6.3% ; 17.2%	12.1% ; 51.4%	
VVS1	Count	17 (0.0%)	186 (0.3%)	789 (1.5%)	616 (1.1%)	2,047 (3.8%)	3,655 (6.8%)
	Mar. pct	1.1% ; 0.5%	3.8% ; 5.1%	6.5% ; 21.6%	4.5% ; 16.9%	9.5% ; 56.0%	
IF	Count	9 (0.0%)	71 (0.1%)	268 (0.5%)	230 (0.4%)	1,212 (2.2%)	1,790 (3.3%)
	Mar. pct	0.6% ; 0.5%	1.4% ; 4.0%	2.2% ; 15.0%	1.7% ; 12.8%	5.6% ; 67.7%	
Total	Count	1,610 (3.0%)	4,906 (9.1%)	12,082 (22.4%)	13,791 (25.6%)	21,551 (40.0%)	53,940 (100.0%)

⁽¹⁾ Columns and rows percentages

Avancées 2023 et « R Tables for Regulatory Submissions Working Group »

The goal of the working group is to create standards for creating tables that meet the requirements of FDA submission documents, and hence enhance the suitability of R for FDA submissions. It is part of a larger R Consortium effort to facilitate the certification and validation of R packages and tools for FDA submissions thereby allowing drug developers to submit documentation for regulatory approval using the R programming environment in conjunction with open-source packages without the need for closed and often expensive proprietary tools.

Deux types de packages :

1. Ceux qui sont centrés sur l'expression fonctionnel, la définition du contenu mais peu de capacité de formatage (tables, rtables, tfrmt, tidytlg)
2. Ceux qui sont centrés sur le formatage, la définition du contenu est centrée sur la mise en forme d'un data.frame (flextable, gt)

	Drug X (N=134)	Placebo (N=134)	Combination (N=132)
Subjects with serious adverse events	89 (66.42%)	123 (91.79%)	88 (66.67%)
Number of censored subjects			
Clinical Cut Off	10 (7.46%)	4 (2.99%)	14 (10.61%)
Completion or Discontinuation	13 (9.70%)	3 (2.24%)	16 (12.12%)
End of AE Reporting Period	22 (16.42%)	4 (2.99%)	14 (10.61%)
Hazard ratio		1.5	1.1
95% confidence interval		(1.1, 1.9)	(0.8, 1.5)
p-value (one-sided stratified log rank)		0.0208	0.4619
Time to first serious adverse event			
Median (years)	0.39	0.37	0.26
95% confidence interval	[0.23, 0.60]	[0.25, 0.46]	[0.18, 0.34]
Min, Max	0.00, 3.00	0.00, 3.00	0.00, 3.00

	A: Drug X (N=134)	B: Placebo (N=134)	C: Combination (N=132)	
Patients with at least one event	122 (91.04%)	123 (91.79%)	120 (90.91%)	Vient d'un autre tableau
Total number of events	609	622	703	Métrique A
cl A.1				
Patients with at least one event	78 (58.21%)	75 (55.97%)	89 (67.42%)	Métrique B
Total number of events	132	130	160	Métrique C
dcd A.1.1.1.1	50 (37.31%)	45 (33.58%)	63 (47.73%)	Métrique D
dcd A.1.1.1.2	48 (35.82%)	48 (35.82%)	50 (37.88%)	Métrique E
cl B.1				
Patients with at least one event	47 (35.07%)	49 (36.57%)	43 (32.58%)	
Total number of events	56	60	62	
dcd B.1.1.1.1	47 (35.07%)	49 (36.57%)	43 (32.58%)	
cl B.2				
Patients with at least one event	79 (58.96%)	74 (55.22%)	85 (64.39%)	
Total number of events	129	138	143	
dcd B.2.1.2.1	49 (36.57%)	44 (32.84%)	52 (39.39%)	
dcd B.2.2.3.1	48 (35.82%)	54 (40.30%)	51 (38.64%)	
....				

6 agrégations différentes à empiler dans un tableau unique et réparties sur 3 groupes

as_flextable()

Le package ‘tables’ n'est pas très loin de la ‘PROC TABULATE’ de SAS et répond à beaucoup de besoins. Le package ‘rtables’ est dédié au reporting clinique et aussi très élégant pour produire des tableaux.

```
> tab <- tabular(  
  (cut + 1) ~ Format(digits=3) *  
    (price + depth) * (mean + sd),  
  data = ggplot2::diamonds )  
> tab
```

cut	price		depth	
	mean	sd	mean	sd
Fair	4358.758	3560.387	64.042	3.643
Good	3928.864	3681.590	62.366	2.169
Very Good	3981.760	3935.862	61.818	1.379
Premium	4584.258	4349.205	61.265	1.159
Ideal	3457.542	3808.401	61.709	0.719
All	3932.800	3989.440	61.749	1.433

Les codes ‘flextable’ ou ‘gt’ pour produire certains tableaux sont **longs car il faut préparer le data.frame**.

Mais l'utilisation de la fonction [as_flextable\(\)](#) permet de s'affranchir de cette inconvénient et offre à ‘tables’ et ‘rtables’ les formats RTF, Word, PowerPoint, HTML, grid graphics, PDF (plus tard Excel)

cut	price		depth	
	mean	sd	mean	sd
Fair	4358.758	3560.387	64.042	3.643
Good	3928.864	3681.590	62.366	2.169
Very Good	3981.760	3935.862	61.818	1.379
Premium	4584.258	4349.205	61.265	1.159
Ideal	3457.542	3808.401	61.709	0.719
All	3932.800	3989.440	61.749	1.433

Adverse Event Table

<https://rconsortium.github.io/rtrs-wg/commontables.html#tables-5>

```
body <- tabular( Heading("Patients with at least one event")*1*  
  Heading("")*countpercentid*Arguments(ARM = ARM)*  
  Heading()*USUBJID +  
  Heading("Total number of events")*1*Heading("")*1 +  
  Heading()*AEBODSYS*  
  (Heading("Patients with at least one event")*  
    Percent(denom = ARM, fn = countpercentid)*  
    Heading()*USUBJID +  
    Heading("Total number of events")*1 +  
    Heading()*AEDECOD*DropEmpty(which = "row")*  
    Heading()*Percent(denom = ARM, fn = countpercentid)*  
    Heading()*USUBJID ~  
    Heading()*ARM, data = ex_adae )  
as_flextable(body, ...)
```

	A: Drug X (N=134)	B: Placebo (N=134)	C: Combination (N=132)
Patients with at least one event	122 (91.04%)	123 (91.79%)	120 (90.91%)
Total number of events	609	622	703
cl A.1			
Patients with at least one event	78 (58.21%)	75 (55.97%)	89 (66.42%)
Total number of events	132	130	160
dcd A.1.1.1.1	50 (37.31%)	45 (33.58%)	63 (47.01%)
dcd A.1.1.1.2	48 (35.82%)	48 (35.82%)	50 (37.31%)
cl B.1			
Patients with at least one event	47 (35.07%)	49 (36.57%)	43 (32.09%)
Total number of events	56	60	62
dcd B.1.1.1.1	47 (35.07%)	49 (36.57%)	43 (32.09%)
cl B.2			
Patients with at least one event	79 (58.96%)	74 (55.22%)	85 (63.43%)
Total number of events	129	138	143
dcd B.2.1.2.1	49 (36.57%)	44 (32.84%)	52 (38.81%)
dcd B.2.2.3.1	48 (35.82%)	54 (40.30%)	51 (38.06%)
cl C.1			
Patients with at least one event	43 (32.09%)	46 (34.33%)	43 (32.09%)
Total number of events	55	63	64
dcd C.1.1.1.3	43 (32.09%)	46 (34.33%)	43 (32.09%)

Time to Event Analysis Table

<https://rconsortium.github.io/rtrs-wg/commontables.html#flextable-6>

	A: Drug X (N=134)	B: Placebo (N=134)	C: Combination (N=132)
Subjects with serious adverse events	89 (66.42%)	123 (91.79%)	88 (66.67%)
Number of censored subjects			
Clinical Cut Off	10 (7.46%)	4 (2.99%)	14 (10.61%)
Completion or Discontinuation	13 (9.70%)	3 (2.24%)	16 (12.12%)
End of AE Reporting Period	22 (16.42%)	4 (2.99%)	14 (10.61%)
Hazard ratio		1.5	1.1
95% confidence interval		(1.1, 1.9)	(0.8, 1.5)
p-value (one-sided stratified log rank)		0.0208	0.4619
Time to first serious adverse event			
Median (years)	0.39	0.37	0.26
95% confidence interval	[0.23, 0.60]	[0.25, 0.46]	[0.18, 0.34]
Min, Max	0.00, 3.00	0.00, 3.00	0.00, 3.00

Concomitant Medications

<https://rconsortium.github.io/rtrs-wg/commontables.html#flextable-7>

Conmed Example

Uses the adcm dataset from **random.cdisc.data**

	A: Drug X	B: Placebo	C: Combination
	(N=134)	(N=134)	(N=132)
At Least One Concomittant Med	122 (91.0%)	123 (91.8%)	120 (90.9%)
medcl A			
A_1/3	54 (40.3%)	49 (36.6%)	69 (52.3%)
A_2/3	53 (39.6%)	50 (37.3%)	56 (42.4%)
A_3/3	45 (33.6%)	54 (40.3%)	48 (36.4%)
medcl B			
B_1/4	52 (38.8%)	57 (42.5%)	59 (44.7%)
B_2/4	52 (38.8%)	55 (41.0%)	56 (42.4%)
B_3/4	47 (35.1%)	47 (35.1%)	52 (39.4%)
B_4/4	50 (37.3%)	45 (33.6%)	55 (41.7%)
medcl C			
C_1/2	51 (38.1%)	50 (37.3%)	56 (42.4%)
C_2/2	52 (38.8%)	58 (43.3%)	60 (45.5%)

Créé avec un objet de rtables cette fois-ci:

```
tt_to_flextable(build_table(lyt, cadcm, alt_counts_df = cads!)) |>
  theme_booktabs() |>
  font(fontname = "Open Sans") |>
  bold(i = ~ V2 %in% "", j = 1, bold = TRUE) |>
  bold(i = 1, j = 1, bold = TRUE) |>
  align(j = 2:4, align = "center", part = "all") |>
  set_table_properties(layout = "fixed") |>
  autofit() |>
  mk_par(i = 1, j = 1, part = "header",
         as_paragraph(as_chunk("Conmed Example",
                               props = fp_text_default(font.size = 14)))) |>
  mk_par(i = 2, j = 1, part = "header",
         as_paragraph("Uses the adcm dataset from ",
                     as_b("random.cdisc.data"))))
```



A JOURNAL OF SPACE
AND TIME IN ECOLOGY



Software notes | Open Access |

NetLogoR: a package to build and run spatially explicit agent-based models in R

Sarah Bauduin Eliot J. B. McIntire, Alex M. Chubaty

First published: 10 August 2019 | <https://doi.org/10.1111/ecog.04516>



= Traduction de NetLogo sur R

Les modèles individu-centrée

Individual-based models (IBM)

Modèles de simulation

Modèles « bottom-up »

Règles de simulation (décisions) à l'échelle de l'individu

Résultats du modèle à l'échelle de la population

Individus = entités uniques et autonomes

Interactions possibles entre individus

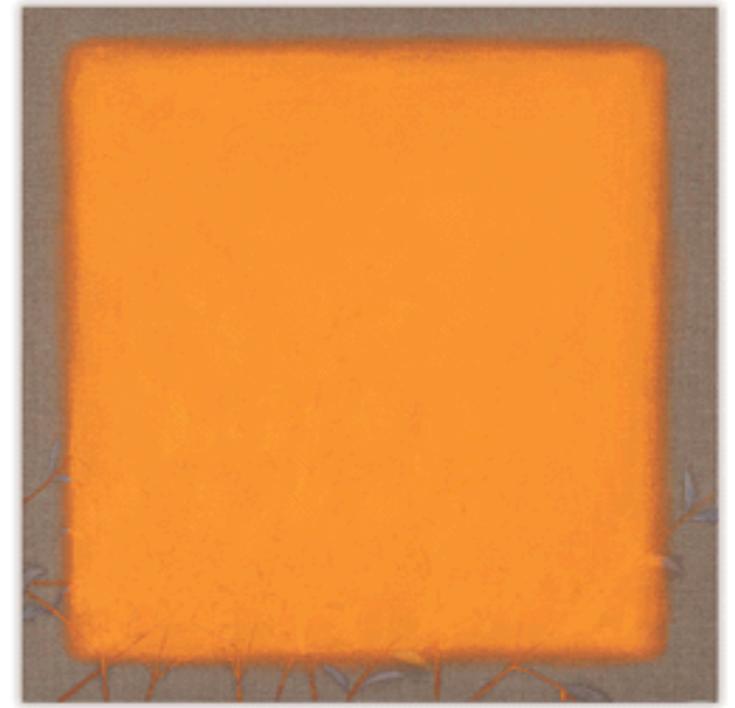
Individus avec comportement adaptable

Modèles individu-centré spatialement explicites (SE-IBM)

Influences/interactions des individus avec l'environnement

Agent-Based and
Individual-Based Modeling

A PRACTICAL INTRODUCTION



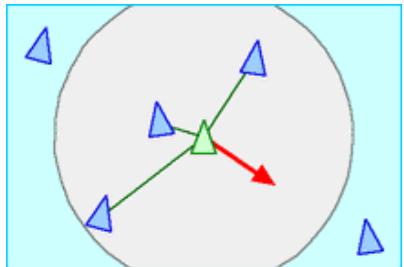
Steven F. Railsback and Volker Grimm

Railsback and Grimm, 2012

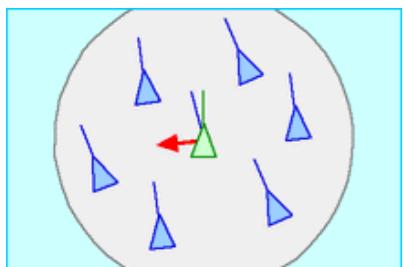
Exemple d'IBM : Envolée d'oiseaux

Reproduire des observations/patrons de populations à partir de règles individuelles simples

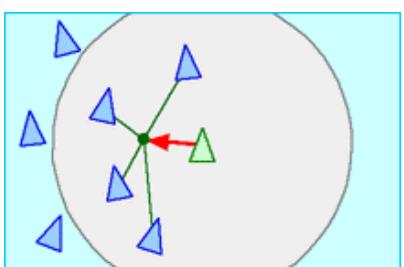
Vol de chaque oiseau = 3 règles simples



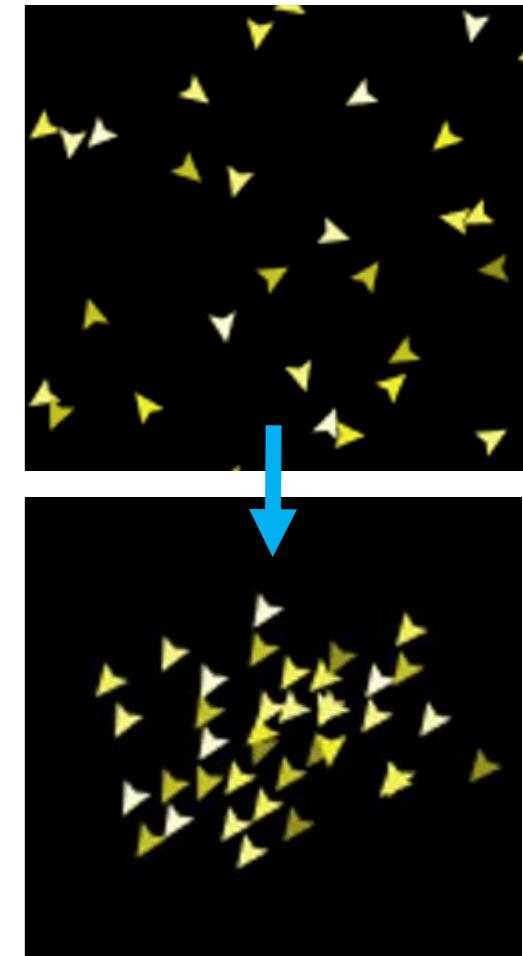
Séparation : s'éloigner des individus trop proches



Alignement : se diriger dans le sens général de la marche



Cohésion : se rapprocher des individus trop éloignés



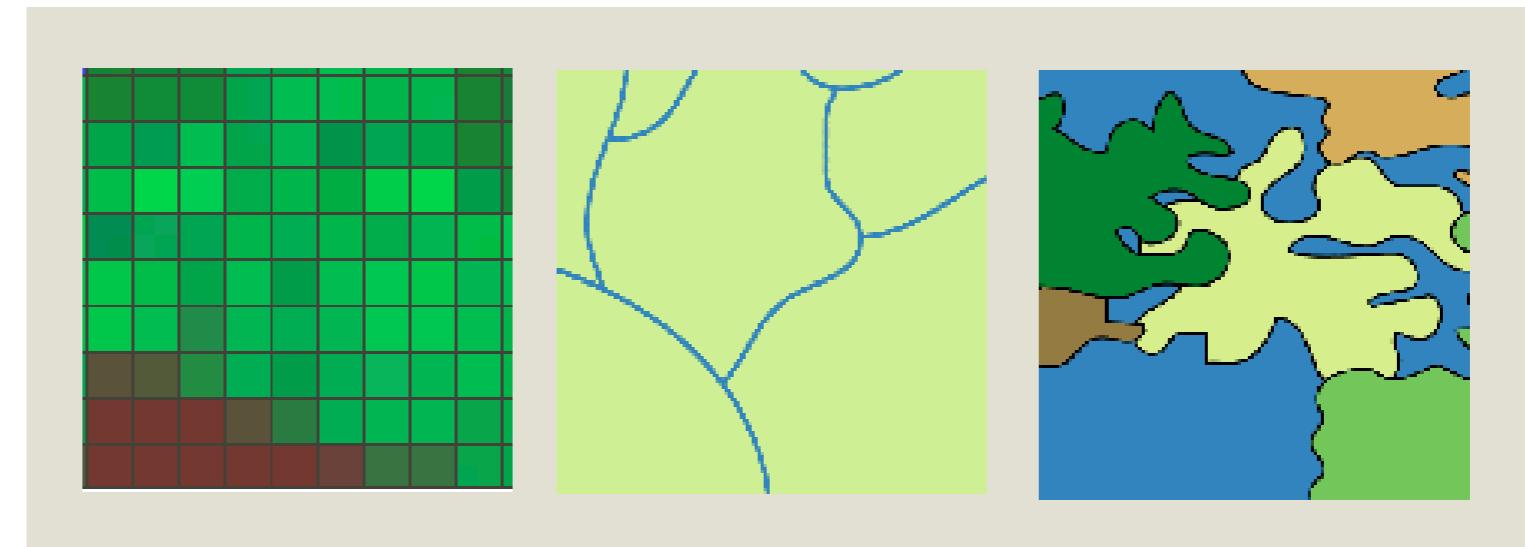
Reynolds, 1986

Avantages de NetLogo

- Structure efficace
- Apprentissage facile et rapide

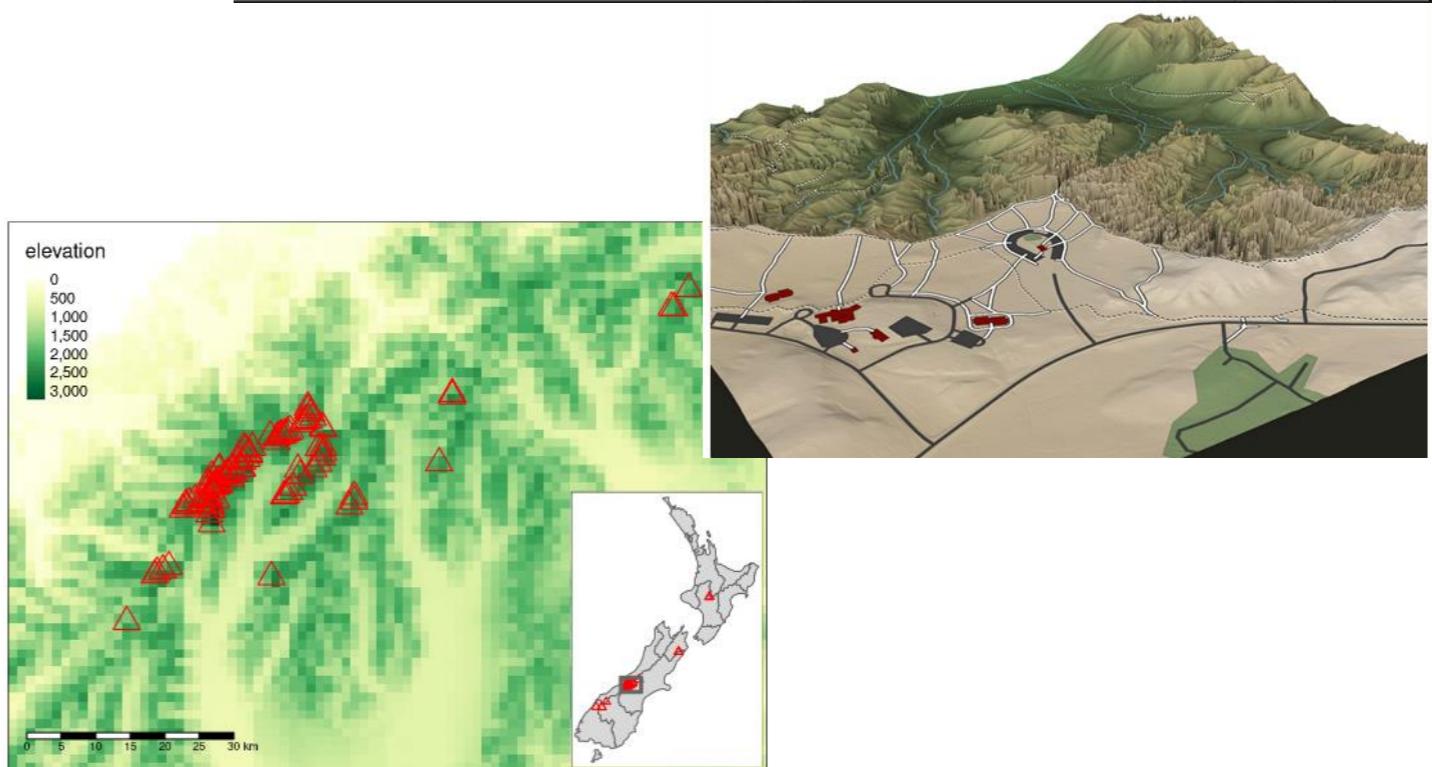
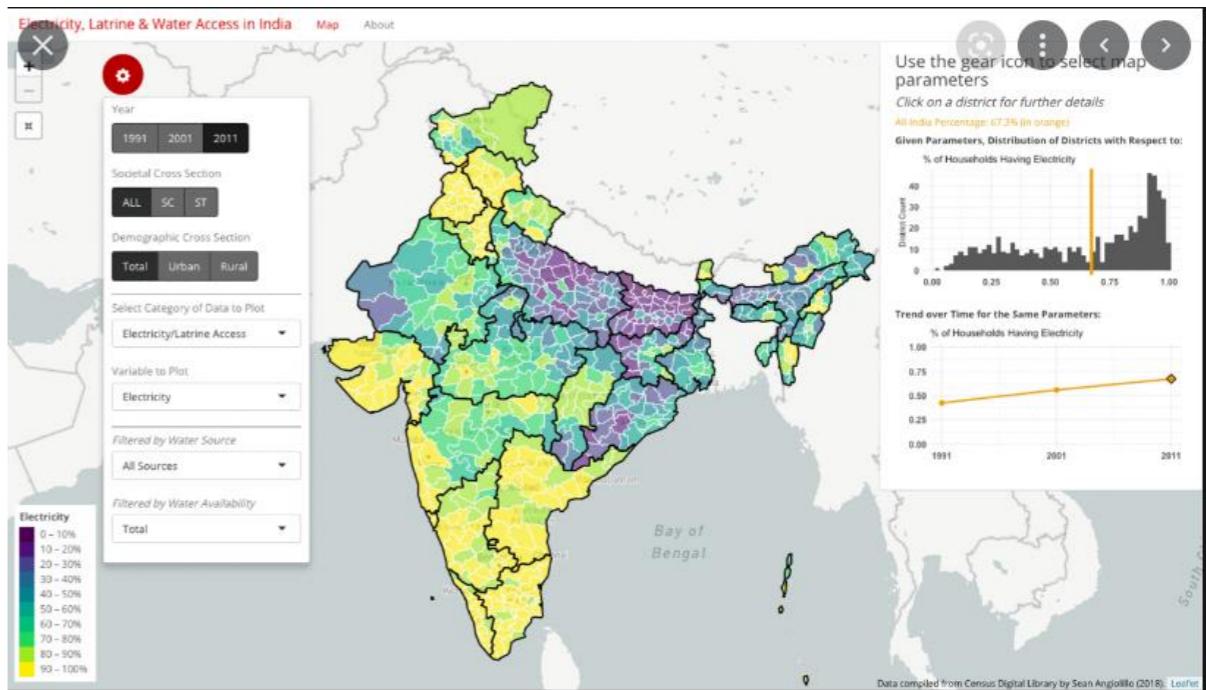
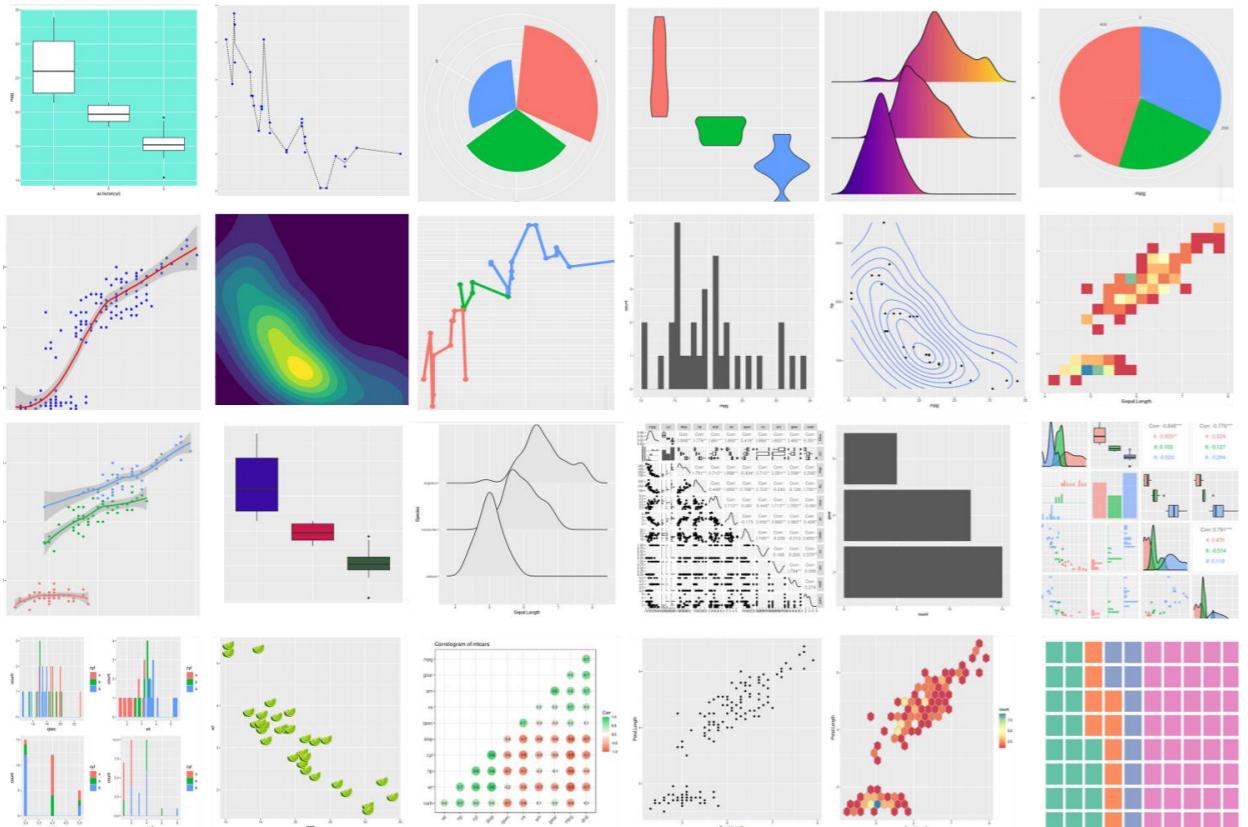
Avantages de R

- Gestion de données (inputs), création de l'IBM, simulations, analyses des résultats (outputs)
= tout sur R !
- Objets de classes différentes :



Avantages de R

- Simulations en parallèle
- Nombreux packages
- Statistiques complexes
- Nombreuses possibilités de visuels



Avantages de R

- Réutilisation de fonctions et paramètres

 **initParam**

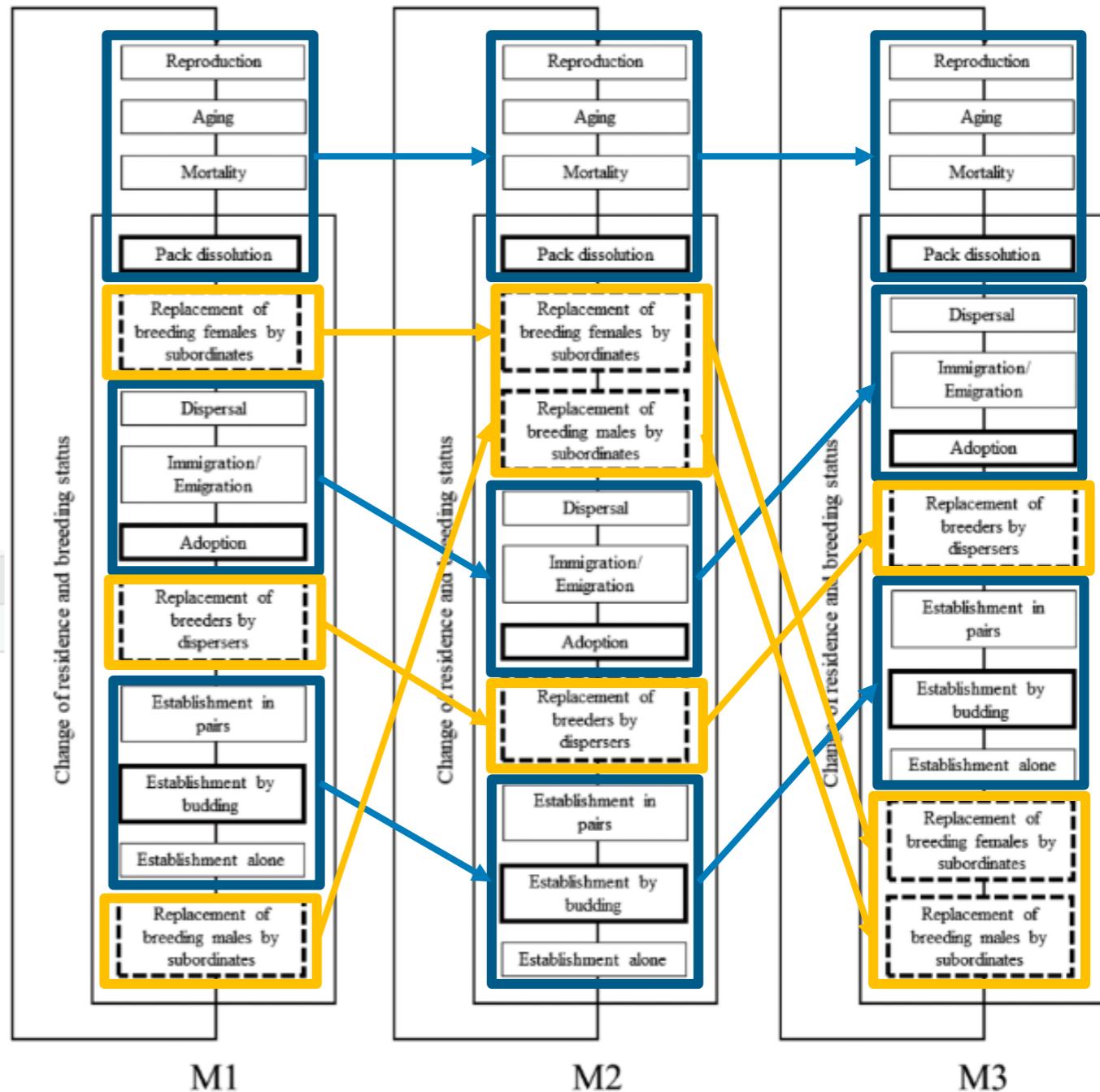
 **run**

 **submodels**

```

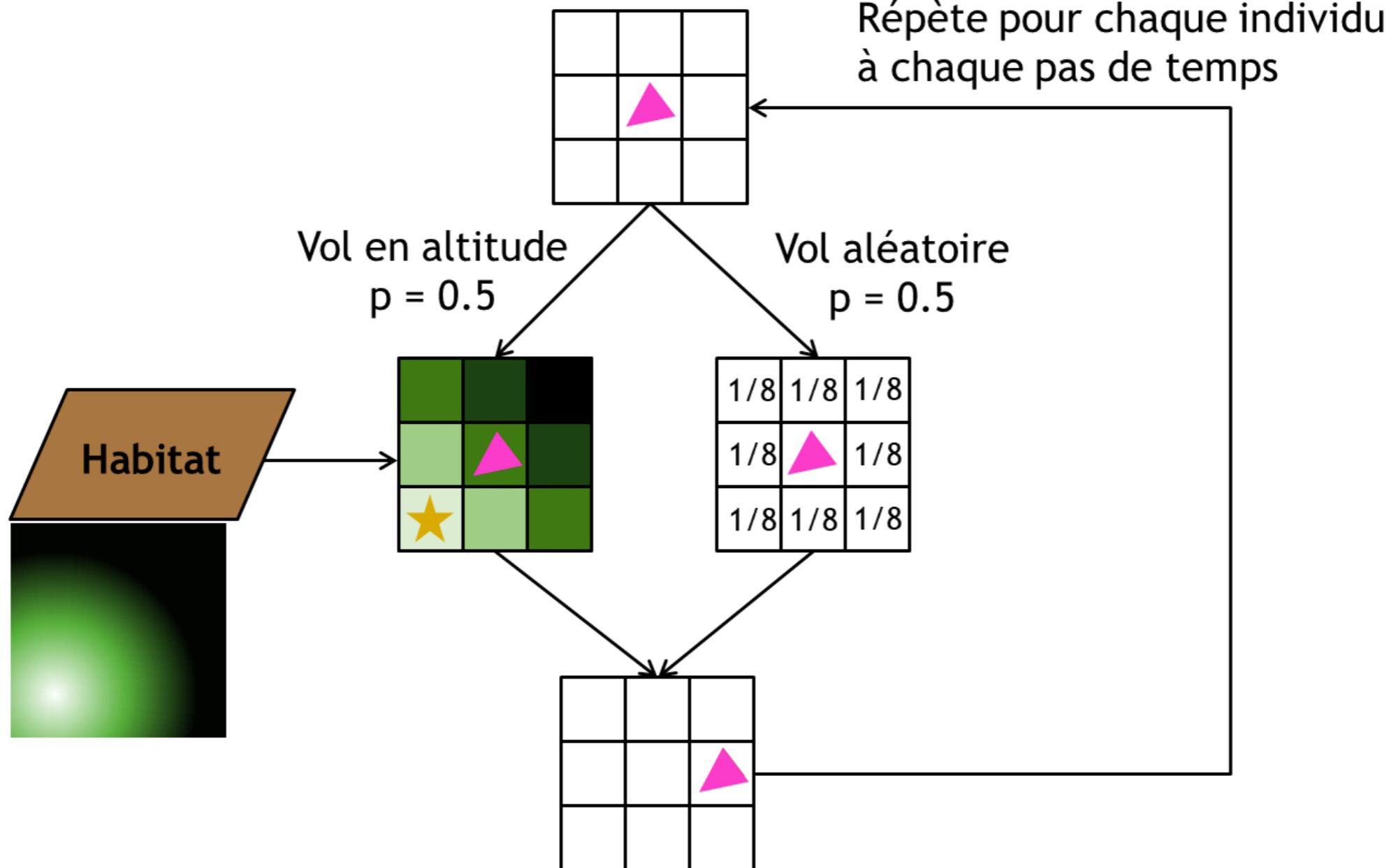
initParam.R * submodels.R * run.R* *
Source on Save | 
  ↗ | ↘ | ↙ | ↖ | ↗ | ↘ | ↙ | ↖ |
  15
  16 - #####
  17 # Prepare the model
  18
  19 # Call the sub-models
  20 source("submodels.R")
  21
  22 # Call the function to init the population and the sub-model
  23 source("initParam.R")
  24
  25 - #####
  26 # Model version M1
  27

```



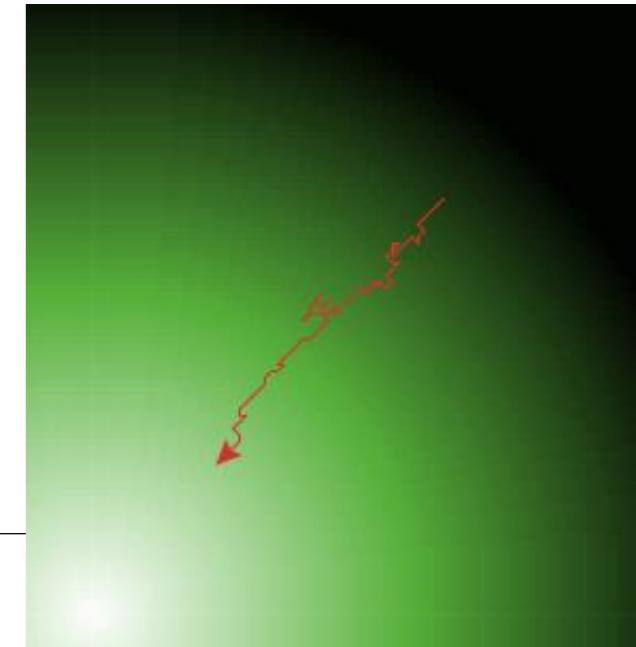
NetLogo function	Description	NetLogo equivalents
bk	Move the individuals backward of their heading's directions	back
canMove	Indicate if an individual can move a given distance without leaving the world's extent	jump can-move?
cellFromPxcorPycor	Report the cell numbers as defined for Raster* object for the given patch coordinates	NA
clearPatches	Reset all patches' value	clear-patches
createOTurtles	Create individuals and place them at the center of the world with their headings evenly distributed	create-ordered-turtles create-ordered<breeds>
createTurtles	Create individuals with a set of defined variables	create-turtles create-<breeds>
createWorld	Create a world of patches	NA
die	Kill selected individuals	die
diffuse	Diffuse values among patches where each patch give an equal share of a portion of its value to its neighboring patches	diffuse diffuse4
downhill	Move the individuals to their neighboring patch with the lowest value	downhill downhill4
dx	Report the amount by which the individual's x-coordinate would change if it would move forward the given distance with its current heading	dx
dy	Report the amount by which the individual's y-coordinate would change if it would move forward the given distance with its current heading	dy
face	Set the individuals' heading towards the given agents	face facexy
fd	Move the individuals forward with their headings as directions	forward jump

Modèle d'un vol de papillons (SE-IBM)



Modèle d'un vol de papillons (SE-IBM)

```
for (time in 1:100) {  
  if (runif(n = 1, min = 0, max = 1) < 0.5) {  
    # Vol en altitude  
    b3 <- uphill(world = hill, turtles = b3, nNeighbors = 8)  
  } else {  
    # Vol aléatoire  
    allNeighbors <- neighbors(world = hill, agents = b3,  
                                nNeighbors = 8)  
    oneNeighbor <- oneOf(allNeighbors)  
    b3 <- moveTo(turtles = b3, agents = oneNeighbor)  
  }  
}
```



Le choix dépend de vos objectifs !

NetLogo

- Rapide et facile à prendre en main

NetLogoR

- Séquence de la gestion des données, la modélisation puis l'analyse des résultats sur R
- Puissance de R

```
install_github("PredictiveEcology/NetLogoR")  
(En mise à jour sur CRAN)
```

Merci de votre attention !
sarah.bauduin@ofb.gouv.fr



Rencontres R 2023

22/06/2023



Combien d'animaux dans mon essai ?

Terence DECHAUX

Institut de l'Elevage – Service Data'Stat

Contexte – L'expérimentation en élevage

L'expérimentation en élevage est un outil essentiel pour optimiser les systèmes de production animale et répondre aux besoins croissants de la population mondiale

Exemple d'essai expérimental

Alimentation A



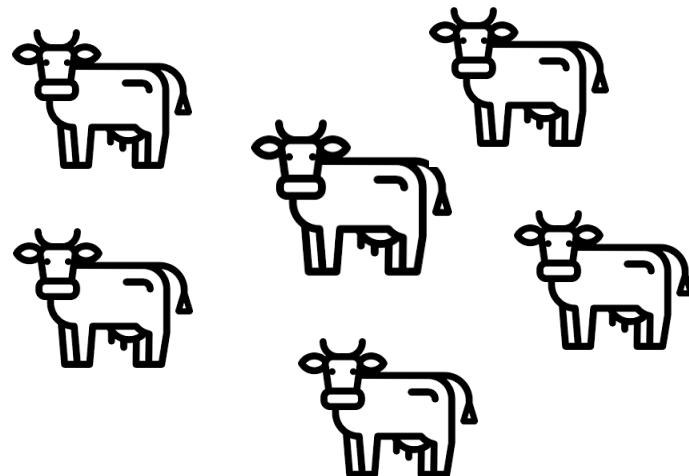
Alimentation B



Production de lait

Contexte – Le protocole expérimental

Pour répondre à cette question, il est indispensable de rédiger un protocole expérimental



Trop d'animaux :

- coûte cher
- Pas toujours de gain de puissance statistique

Pas assez d'animaux :

- Pas assez de puissance statistique
- Impacte sur les résultats du traitement statistique

Idée de la création d'un outil Shiny

Comment **accompagner** efficacement nos collègues dans le dimensionnement de leurs essais pour en faire une réussite ?



**Une application Shiny permettant de réaliser
soit même le calcul d'effectif**

L'application CalculEffectifs

L'application Shiny CalculEffectifs a été développée dans le but d'accompagner les équipes métier dans l'élaboration des protocoles expérimentaux

Calcul de taille d'échantillon

SÉLECTIONNER LE TEST STATISTIQUE

Comparaison de 2 lots

Seuil de significativité

10%

5%

2%

Puissance

60%

80%
100%

Type de test

Bilatéral

Unilatéral

Déifference de moyennes attendue

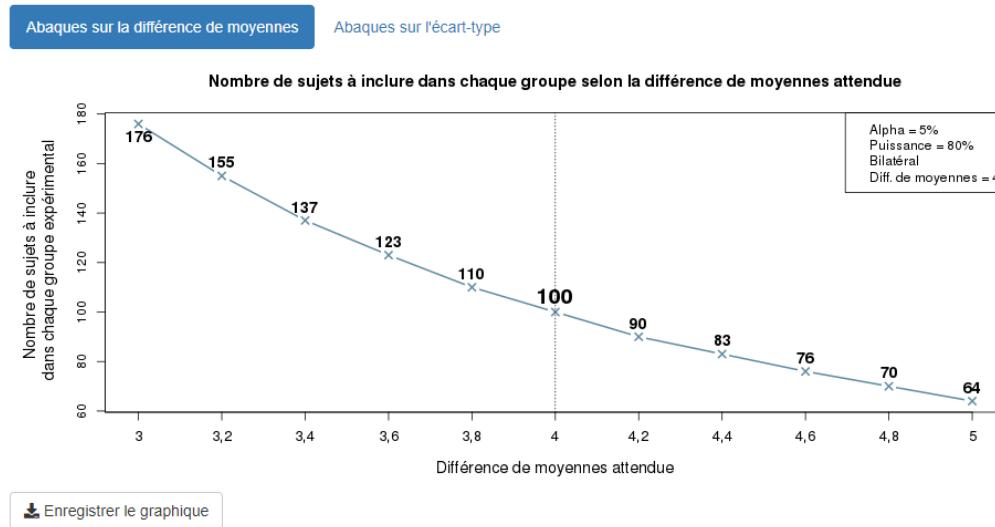
Ecart-type attendu

Intervalle du graphique des abaques

Min	Max	Pas
3	5	0,2

Intervalle du graphique des abaques

Min	Max	Pas
8,5	11,5	0,3



Fonctionnement de l'application Shiny – Paramétrage

SÉLECTIONNER LE TEST STATISTIQUE

Comparaison de 2 lots

Choix du dispositif

Seuil de significativité

10%
 5%
 2%

Puissance

60% 80% 100%

60 65 70 75 80 85 90 95 100

Type de test

Bilatéral
 Unilatéral

Déférence de moyennes attendue

4

Choix du risque de conclure à tort à un effet

Ecart-type attendu

10

Choix d'un test de différence ou de supériorité

La définition des limites des axes des abaques

Min	Max	Pas
3	5	0,2

La définition des limites des axes des abaques

Min	Max	Pas
8,5	11,5	0,3

Choix de la puissance (i.e. la faculté de mettre en évidence une différence qui existe)

Les paramètres statistiques (ne pas modifier inconsciemment les valeurs par défaut)

Déférence de moyennes attendue

4

Intervalle du graphique des abaques

Min

3

Max

5

Pas

0,2

Ecart-type attendu

10

Intervalle du graphique des abaques

Min

8,5

Max

11,5

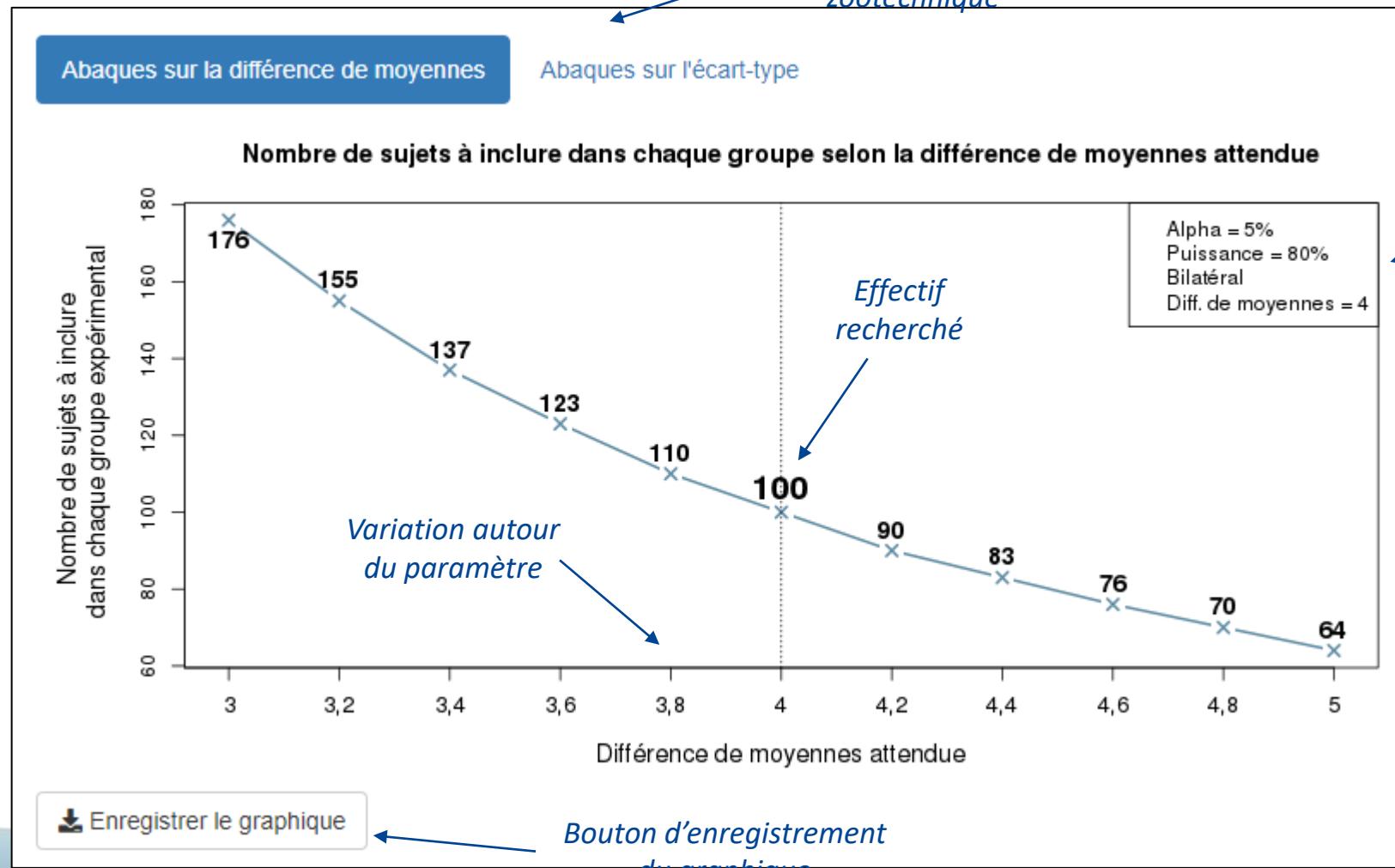
Pas

0,3

Les paramètres zootechniques

- Différence attendue
- Ecart-type attendu

Fonctionnement de l'application Shiny – Abaques



Fonctionnement de l'application Shiny – Blocs

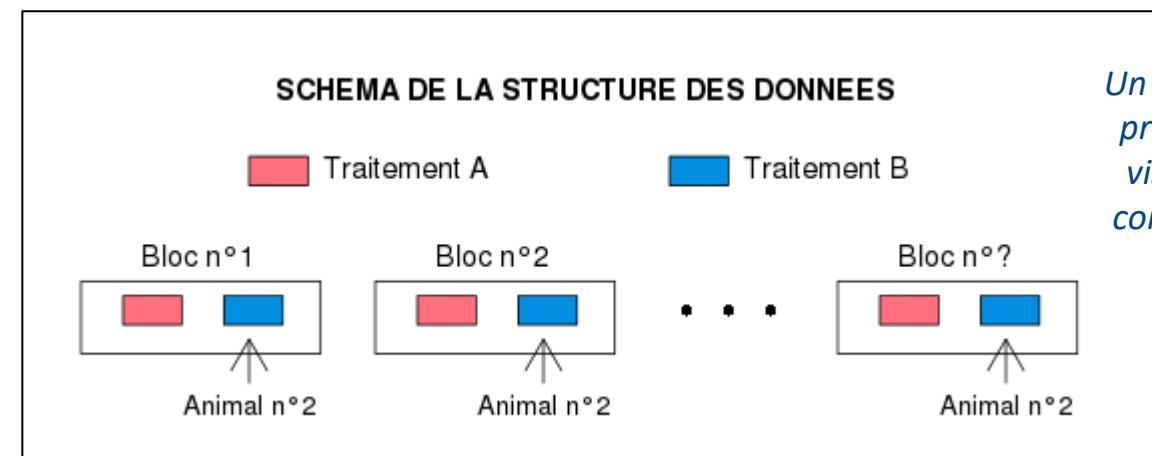
*Corrélation des mesures d'un bloc
(à estimer à partir d'anciennes expés)*

STRUCTURE DE CORRÉLATION Aidez-vous du schéma à droite pour choisir les paramètres.

Corrélation des mesures dans un bloc	Intervalle du graphique des abaques
0,1	<input max="1.0" min="0.05" type="range" value="0.2"/>
Effectif recherché	Nombre d'animaux par bloc
<input checked="" type="radio"/> Nombre de blocs <input type="radio"/> Nombre d'animaux par bloc	2

*L'effectif recherché
(en général, le nombre de blocs)*

La définition des limites des axes des abaques



Un schéma est produit pour visualiser la configuration

Prise en compte des contraintes terrains

S'il existe des contraintes terrains qu'on doit prendre en compte, il convient de l'intégrer au calcul d'effectifs

SÉLECTIONNER LE TEST STATISTIQUE

- Comparaison de 2 lots avec case
- Expérimentation comparant 2 moyennes
- Comparaison de 2 lots
- Comparaison de 2 lots avec bloc
- Comparaison de 2 lots avec case**
- Expérimentation comparant 3 ou 4 moyennes
- Comparaison de 3 ou 4 lots sur des moyennes

SCHEMA DE LA STRUCTURE DES DONNEES

Legend: Traitement A (Red), Traitement B (Blue)

Elevage	Animal	Traitement
n°1	1	A
n°1	2	A
n°1	3	A
n°1	4	A
n°1	5	A
n°1	6	A
n°1	7	A
n°1	8	A
n°1	9	A
n°1	10	A
n°2	1	B
n°2	2	B
n°2	3	B
n°2	4	B
n°2	5	B
n°2	6	B
n°2	7	B
n°2	8	B
n°2	9	B
n°2	10	B
n°?	1	A
n°?	2	A
n°?	3	A
n°?	4	A
n°?	5	A
n°?	6	A
n°?	7	A
n°?	8	A
n°?	9	A
n°?	10	A

SÉLECTIONNER LE TEST STATISTIQUE

- avec la contrainte d'un seul lot par élevage
- Experimentation comparant 2 proportions
- Comparaison de 2 lots sur des proportions
- Expérimentation dans plusieurs élevages avec la contrainte d'un seul lot par élevage**
- avec la possibilité d'avoir les deux lots dans chaque élevage

Conclusion

Pour accéder à l'application : <http://appsonline.idele.fr/CalculEffectif/>

Appeler un statisticien après que l'expérience est terminée, c'est comme lui demander de faire une autopsie ; il pourra seulement déterminer la cause de l'échec de l'expérience.

Sir Ronald A. Fisher, FRS

Suivez nous sur Twitter et LinkedIn



{happign}



Une porte ouverte sur les données IGN

R

IGN

happign, pourquoi ?

PROBLEMES :

Passage par du téléchargement direct :

- Source de données pas toujours facile à trouver ;
- Paquets de données parfois non cohérents avec les besoins ;
- Mise à jour fastidieuse

happign, pourquoi ?

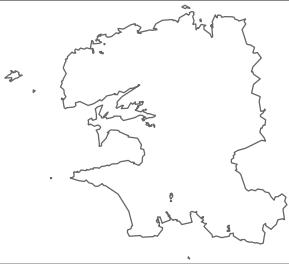
BESOINS :

Être plus agile avec l'utilisation des données :

- Besoin de comparer facilement différentes données ;
- Besoin d'optimisation ;
- Besoins de reproductibilité

happign, comment ?

(WMS et WFS)

Zone Géographique	Type de flux	Catégories	Nom de couche
	WFS (vecteur) WMS (raster)	<ul style="list-style-type: none">• Administratif• Altimétrie• Environnement• Ortho...	<i>LIMITES_ADMINISTRATIVES_EXPRESS.LATEST:canton</i>
Package sf	<code>get_wfs()</code> <code>get_wms_raster()</code>	<code>get_apikeys()</code>	<code>get_layers_metadata()</code>

happign, comment ?

(WMS et WFS)

Récupération
des
catégories

```
> get_apikeys()

c("administratif", "adresse", "agriculture", "altimetrie", "cartes",
  "cartovecto", "clc", "economie", "environnement", "geodesie",
  "lambert93", "ocsg", "ortho", "orthohisto", "parcellaire",
  "satellite", "sol", "topographie", "transports")
```

Récupération
des
noms
de
couche

```
> get_layers_metadata("topographie", "wfs")
```

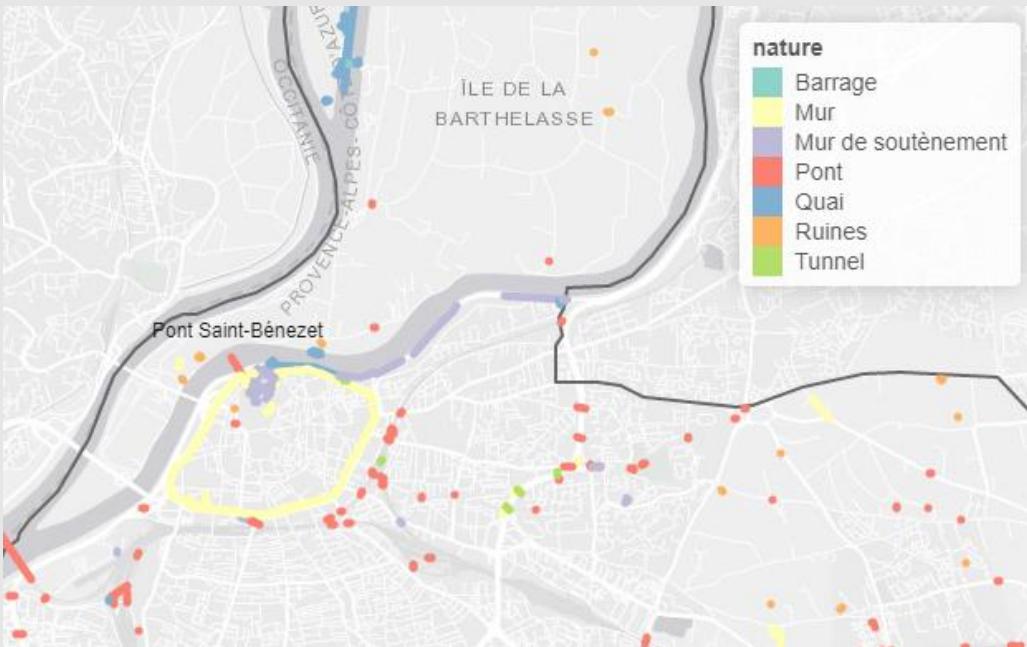
Name	Title	Abstract	...
BDTOPO_V3:construction_lineaire	BDTOPO_V3:construction_lineaire.title	BDTOPO V3 22.4 édition 2022	...
BDTOPO_V3:construction_ponctuelle	BDTOPO_V3:construction_ponctuelle.title	BDTOPO V3 22.4 édition 2022	...
BDTOPO_V3:construction_surfacique	BDTOPO_V3:construction_surfacique.title	BDTOPO V3 22.4 édition 2022	...
...

happign, comment ?

(WFS)

Récupération
des données

```
> get_wfs(shape = read_sf("avignon.shp"),
           apikey = "topographie",
           layer_name = "BDTOPO_V3:construction_lineaire")
```



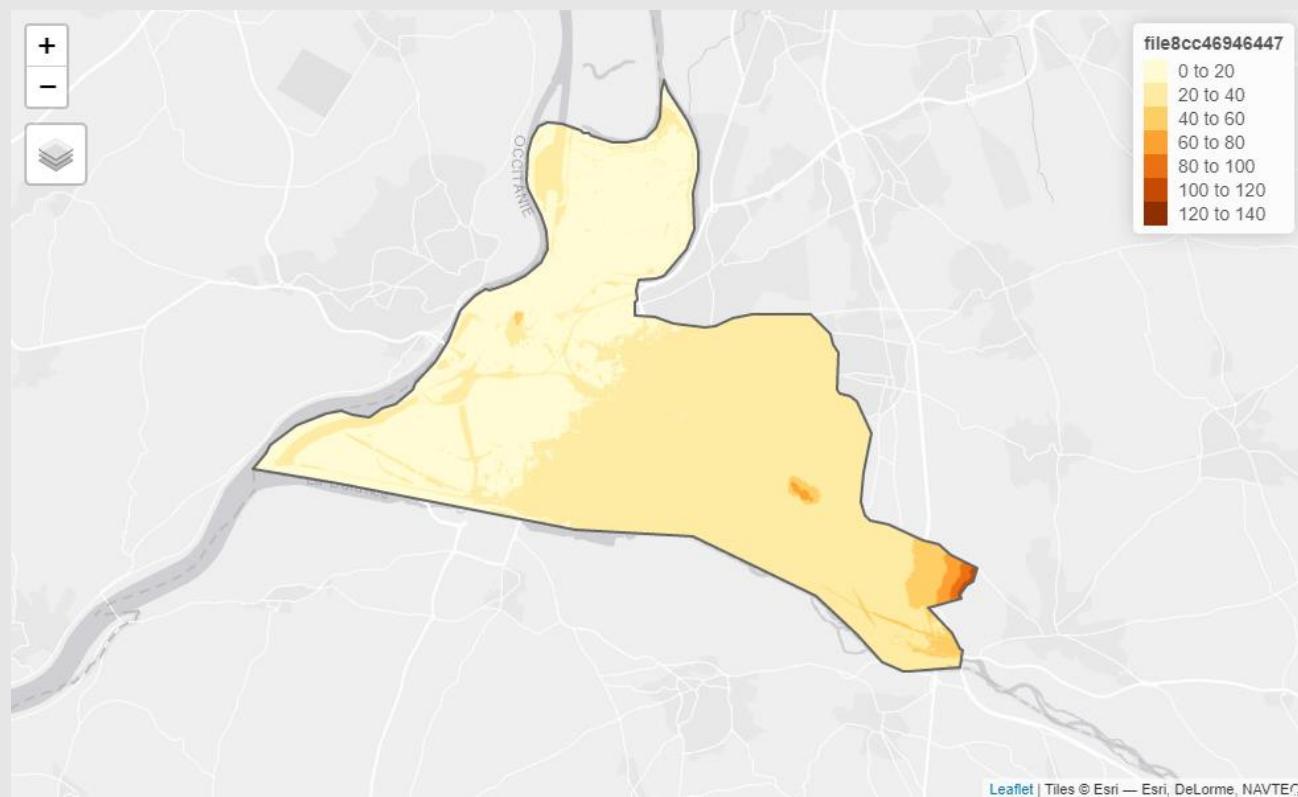
nature	nature_detaillee	toponyme	statut_du_toponyme	importance	etat_de_l_objet	...
Pont	Pont isolé	Pont Saint-Bénézet	Validé	3	En service	...
Mur	Rempart	NA	NA	3	En service	...
Mur	NA	NA	NA	5	En service	...
...

happign, comment ?

(WMS)

Récupération
des données

```
➤ get_wms_raster(shape = read_sf("avignon.shp"),  
                  apikey = "altimetrie ",  
                  layer_name = "ELEVATION.ELEVATIONGRIDCOVERAGE.HIGHRES",  
                  resolution = 5)
```

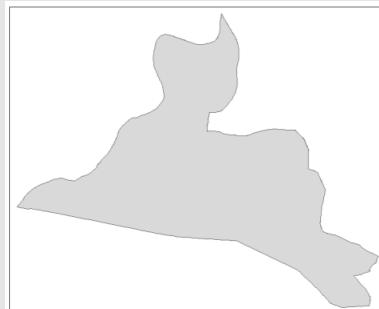


happign, autres sources ?

API Carto : Ne nécessite pas forcément une entrée géographique !

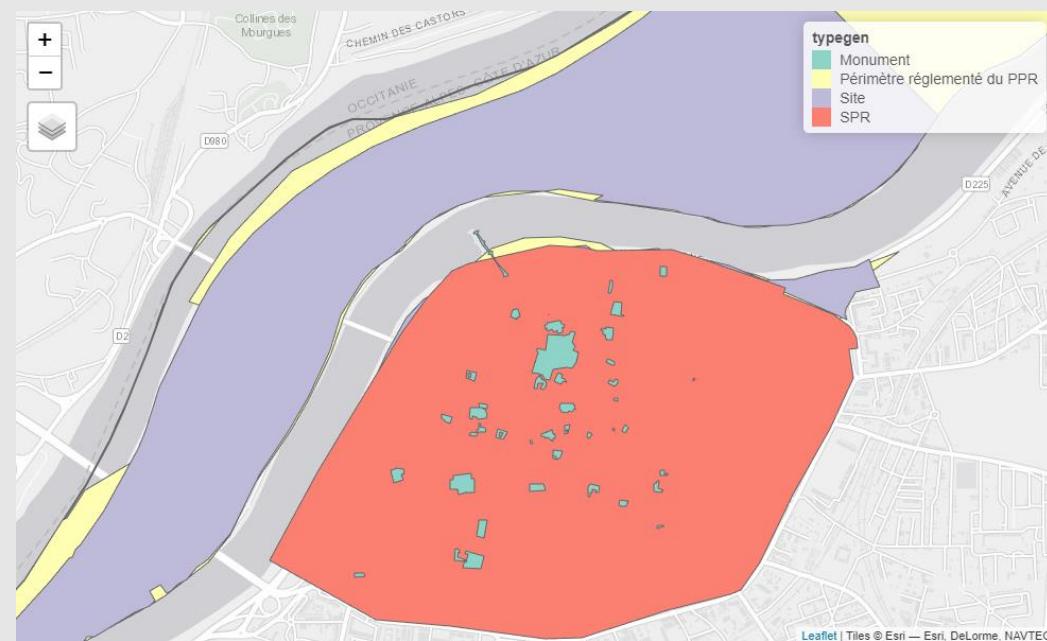
get_apicarto_cadastre

- Parcelles cadastrales
- Communes
- Feuille
- Division
- localisant



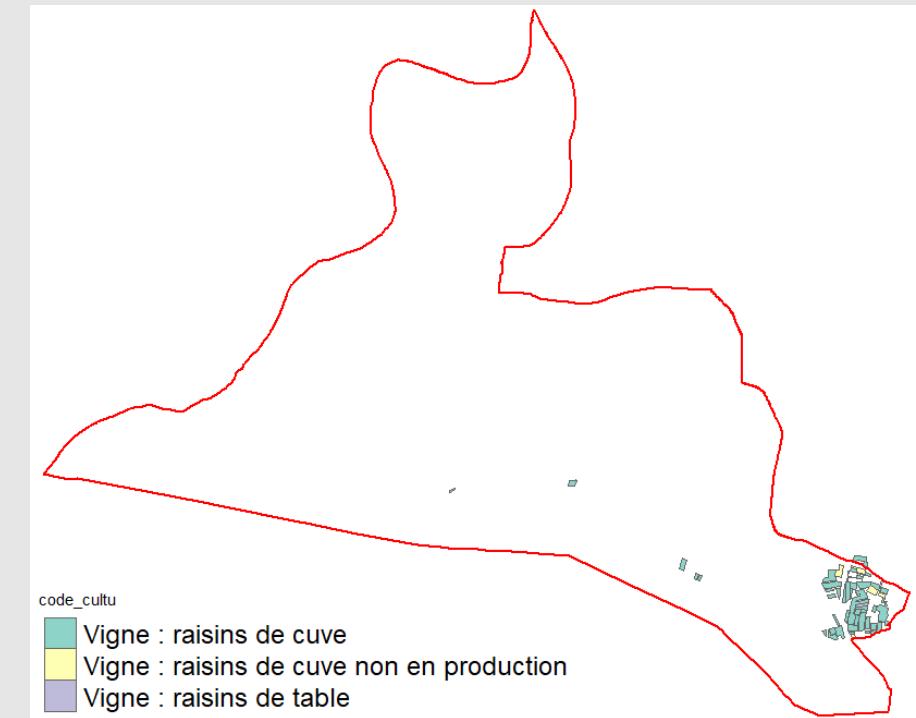
get_apicarto_gpu

- Prescription
- Servitude d'utilité publique



get_apicarto_rpg

- Culture par parcelles par année

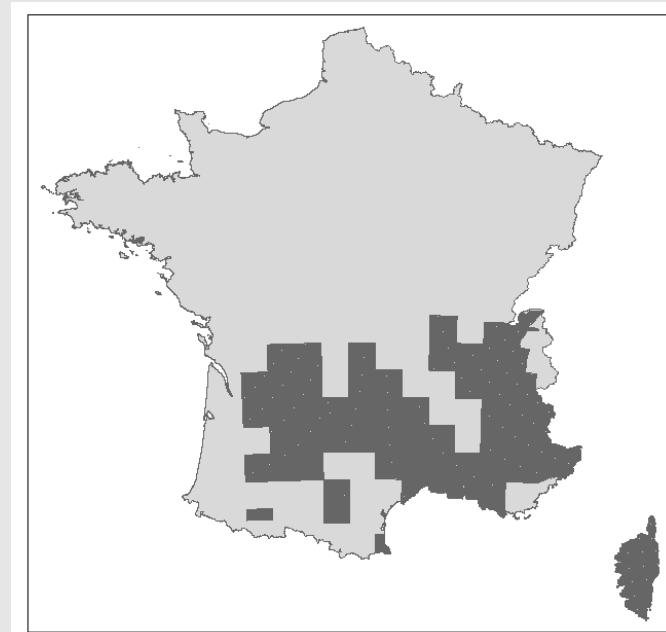


happign, autres sources ?

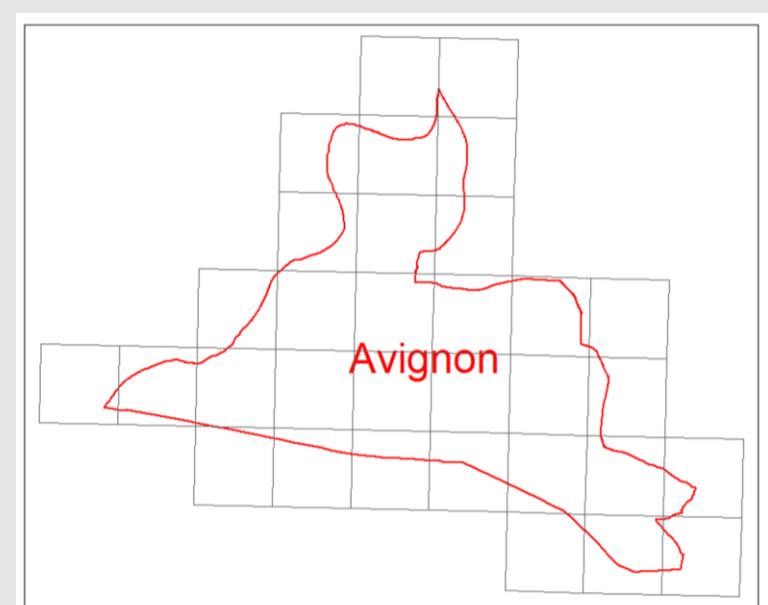
Les données LIDAR brutes (.laz) peuvent être récupérées avec la fonction `get_raw_lidar()`. Elle télécharge la grille d'assemblage (<https://pcrs.ign.fr/download/lidar/shp>), réalise une intersection avec la shape et télécharge les dalles disponibles. Que ce soit la grille d'assemblage ou les dalles .laz, la fonction vérifie si les fichiers ne sont pas déjà téléchargés.

```
➤ get_raw_lidar(shape = get_apicarto_cadastre("84007", "commune"),  
                 destfile = ".",  
                 grid_path = ".",  
                 quiet = F)
```

Grille d'assemblage



Intersection



Liens utiles

Site happign : <https://paul-carteron.github.io/happign/index.html>

Répertoire github : <https://github.com/paul-carteron/happign>

Géoservice web expert : <https://geoservices.ign.fr/services-web-experts>

Documentation géoservice : <https://geoservices.ign.fr/documentation>

Documentation API carto : <https://apicarto.ign.fr/api/doc/>

Merci pour votre attention !

survivalGPU : Analyses de survie sur cartes graphiques

Alexis van STRAATEN¹ Jean FEYDY² Anne-Sophie JANNOT^{2,3}

¹Assistance Publique-Hôpitaux de Paris (AP-HP), Service d'informatique Médicale, Biostatistiques Et Santé Publique,
Hôpital Européen Georges Pompidou, Paris

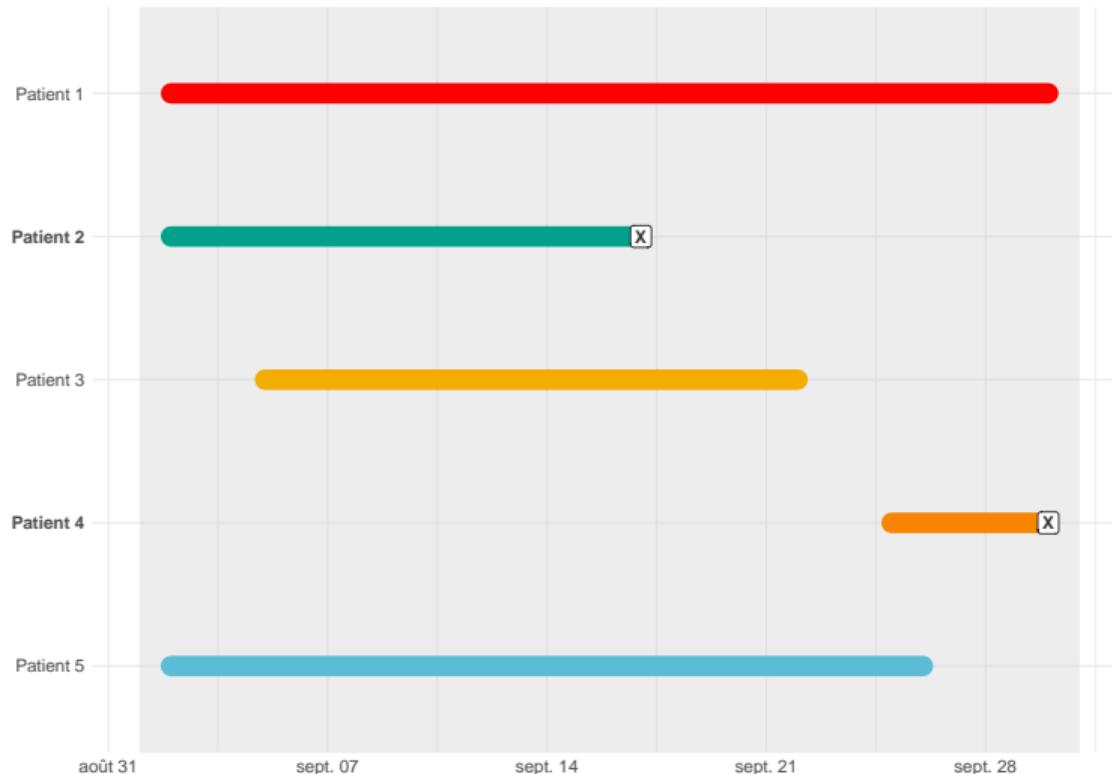
²HeKA, Centre de Recherche des Cordeliers, INSERM, INRIA, Paris

³Cellule opérationnelle de la Banque Nationale de Données Maladies Rares, AP-HP, Paris



Contexte des données censurées

Données avec une fenêtre d'observation qui dépend du patient.



L'analyse de survie

Modèle de régression avec données censurées : Cox (1972)

Implémentation sur R : package survival de Terry Therneau, avec la fonction coxph()

Actuellement dans la recherche médicale:

- ▶ Accès à de larges bases de données de santé.
- ▶ survival est acceptable pour des petites études, mais non utilisable pour des études à grande échelle.

Solution : utiliser les ressources des cartes graphiques.

Objectif : package survival avec support GPU.

survivalGPU

Implémentation de 2 modèles :

- ▶ Cox
- ▶ WCE (Weighted Cumulative Exposure) pour la pharmacovigilance

Plus-values de survivalGPU :

- ▶ Scalable
- ▶ Rapide
- ▶ Gestion des bootstraps
- ▶ Rétro-compatibilité avec survival



Construction de survivalGPU

Etapes de construction du package :

1. Revue du code du package survival
2. Traduction du solveur de Cox (Newton sur fonction convexe) en python avec pytorch pour la compatibilité avec les cartes graphiques Nvidia
3. Construction du package R avec les fonctions python en utilisant reticulate
4. Tests unitaires : reprise des tests de survival et résultats identiques entre survival et survivalGPU (en cours)

Tests unitaires

Exemple d'un test unitaire dans survival

```
test3 <- data.frame(futime=1:12,
                     fustat=c(1,0,1,0,1,0,0,0,0,0,0,0),
                     x1=rep(0:1,6),
                     x2=c(rep(0,6), rep(1,6)))
test3
futime fustat x1 x2
1      1       1  0  0
2      2       0  1  0
3      3       1  0  0
4      4       0  1  0
5      5       1  0  0
6      6       0  1  0
7      7       0  0  1
8      8       0  1  1
9      9       0  0  1
10    10      0  1  1
11    11      0  0  1
12    12      0  1  1

library(survival)
fit3 <- coxph(Surv(futime, fustat) ~ x1 + x2, test3, iter=25)

# Actual solution
# time 1, 12 at risk, 3 each of x1/x2 = 00, 01, 10, 11
# time 2, 10 at risk,           2, 3, 2 , 3
# time 5, 8 at risk,          1, 3, 1, 3
# Let r1 = exp(beta1), r2= exp(beta2)
# loglik = -log(3 + 3r1 + 3r2 + 3 r1*r2) - log(2 + 2r1 + 3r2 + 3 r1*r2) -
#           log(1 + r1 + 3r2 + 3 r1*r2)

true <- function(beta) {
  r1 <- exp(beta[1])
  r2 <- exp(beta[2])
  loglik <- -log(3*(1+ r1+ r2+ r1*r2)) - log(2+ 2*r1 + 3*r2 + 3*r1*r2) -
    log(1 + r1 + 3*r2 + 3*r1*r2)
  loglik
}

paste("log vraisemblance avec coxph() :", round(fit3$loglik[2], 3))
[1] "log vraisemblance avec coxph() : -1.792"
paste("log vraisemblance avec true()  :", round(true(fit3$coef), 3))
[1] "log vraisemblance avec true()  : -1.792"
```

Applications avec le jeu de données synthétique drugdata

Exemples et tests avec WCE::drugdata

```
library(WCE)
drugdata[43:53,]
```

	Id	Event	Start	Stop	sex	age	dose
43	1	0	42	43	1	40	2.5
44	1	0	43	44	1	40	2.5
45	1	0	44	45	1	40	2.5
46	1	0	45	46	1	40	2.5
47	1	0	46	47	1	40	2.5
48	1	0	47	48	1	40	2.5
49	1	1	48	49	1	40	2.5
366	2	0	0	1	0	48	0.0
367	2	0	1	2	0	48	0.0
368	2	0	2	3	0	48	0.0
369	2	0	3	4	0	48	0.0

Description drugdata		
Nombre de données		77 038
Nombre de patients		500
Evenements		383 (77%)
Sexe	Hommes	365 (73%)
	Femmes	135 (27%)
Age		40.76 (13.13)
Suivi		154.08 (101.11)

mean (sd) for numeric variables
N (%) for categorical variables

Comparaison survival et survivalGPU

```
library(survival)

## Cox model
coxph_model <- coxph(Surv(Start, Stop, Event) ~ dose,
                      data = drugdata)
round(exp(coxph_model$coefficients), 3) # HR (Hazard Ratio)

dose
1.246
```

```
## WCE model
wce_model <- WCE(drugdata, "Cox", nknots = 1, cutoff = 90, id = "Id",
                   event = "Event", start = "Start", stop = "Stop",
                   expos = "dose", covariates = c("sex", "age"))
summary(wce_model)
```

Unconstrained estimated WCE function (Proportional hazards model).***

Estimated coefficients for the covariates:

	coef	exp(coef)	se(coef)	z	p
sex	0.6876	1.9889	0.1189	5.7849	0.0000
age	0.0116	1.0116	0.0040	2.9097	0.0036

Partial log-likelihood: -1891.644 BIC: 3824.924

Number of events: 383

Use plot(wce_model) to see the estimated weight function corresponding to this model.

If you report these results, please cite Sylvestre MP, Abrahamowicz M. Flexible Modeling of the Effects of Time-Dependent Exposures on the Hazard. Statistics in Medicine 2009; 28(27):3437-3453.

```
library(survivalGPU)

## Cox model
coxphGPU_model <- coxphGPU(Surv(Start, Stop, Event) ~ dose,
                             data = drugdata)
round(exp(coxphGPU_model$coefficients), 3)

dose
[1,] 1.246
```

```
## WCE model
wceGPU_model <- wceGPU(drugdata, nknots = 1, cutoff = 90, id = "Id",
                        event = "Event", start = "Start", stop = "Stop",
                        expos = "dose", covariates = c("sex", "age"))
summary(wceGPU_model)
```

Estimated coefficients for the covariates :

	coef	CI 2.5 %	CI 97.5 %	exp(coef)	se(coef)	z	p
sex	0.6876	0.4546	0.9206	1.9889	0.1189	5.8	7e-09 ***
age	0.0116	0.0038	0.0194	1.0116	0.0040	2.9	0.004 **

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

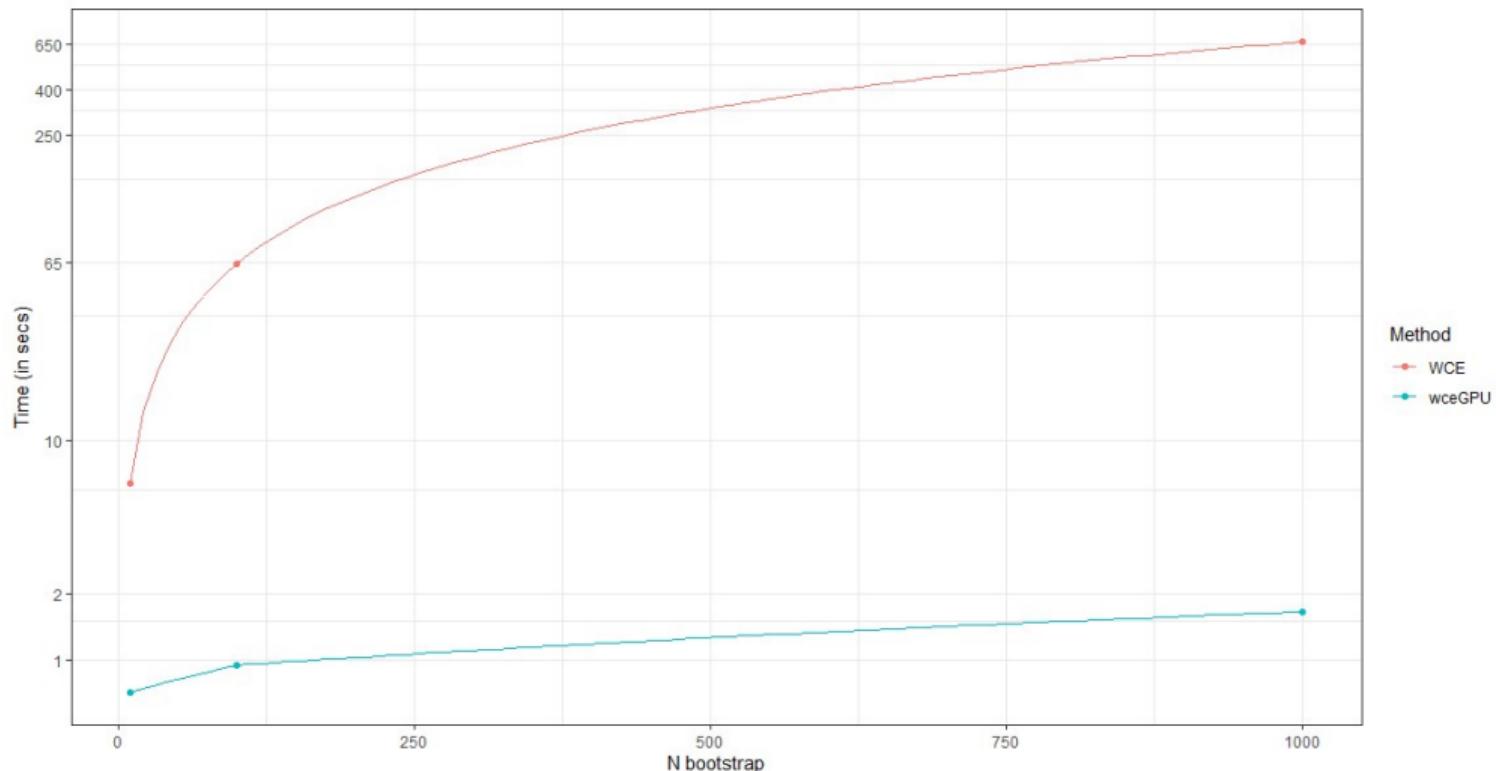
Number of events : 383

Partial log-Likelihoods : -1891.64

BIC : 3824.92

Benchmark du modèle WCE

Benchmark du modèle WCE avec drugdata



Perspectives

- ▶ Implémentation des features de survival, et de nouvelles options tels que les tests de permutation
- ▶ Validation du package à travers des simulations intensives

Package disponible sur Github : <https://github.com/jeanfeydy/survivalGPU>