

RUSS 05-2023

esquisse

Une application pour visualiser ses données avec ggplot2



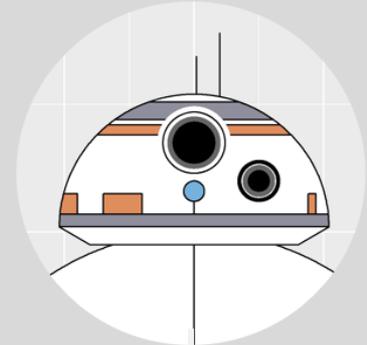
Qui suis-je

Victor



Tavaille à dreamRs

Développe
principalement des
packages autour de
{shiny}



Twitter:

https://twitter.com/_pvictor

GitHub:

<https://github.com/pvictor>

esquisse



**Moins tu sais coder,
plus tu vas rigoler !**



- Présentation du paquet et de ses fonctionnalités
- Des modules et encore des modules
- Construction d'appels à des fonctions avec `{rlang}`
- Internationalisation
- Les Addins Rstudio

esquisse



Présentation du paquet et de ses fonctionnalités

esquisse

C'est quoi ? Pour qui ?



Un paquet R contenant une extension RStudio sous forme d'application {shiny} permettant de créer des graphiques avec {ggplot2}



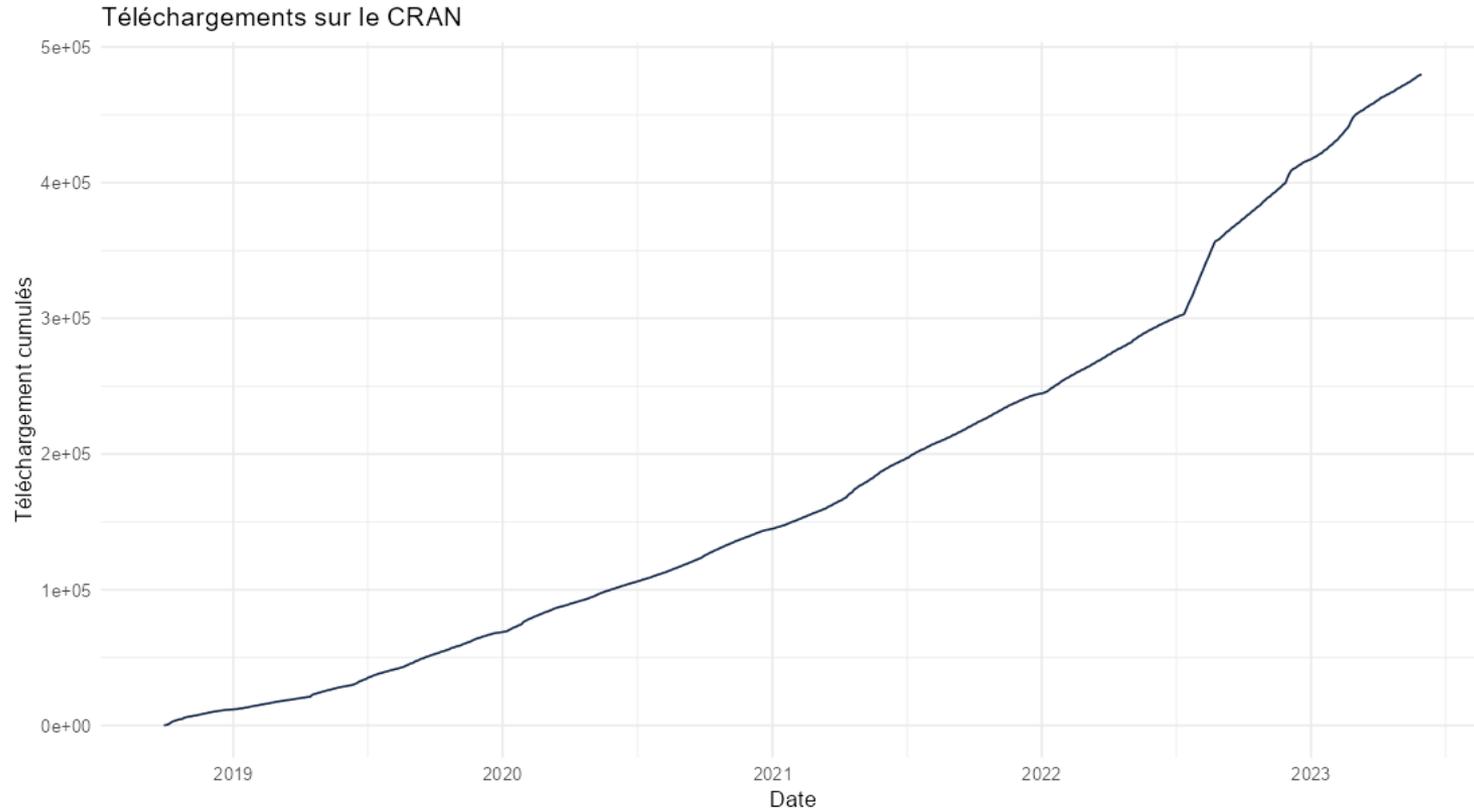
- Débutants en ggplot2
- Utilisateurs occasionnels
- Pour explorer graphiquement des données

esquisse

CRAN



Disponible sur le CRAN depuis septembre 2018, aujourd'hui bientôt un demi-million de téléchargements.



Données : cranlogs

esquisse

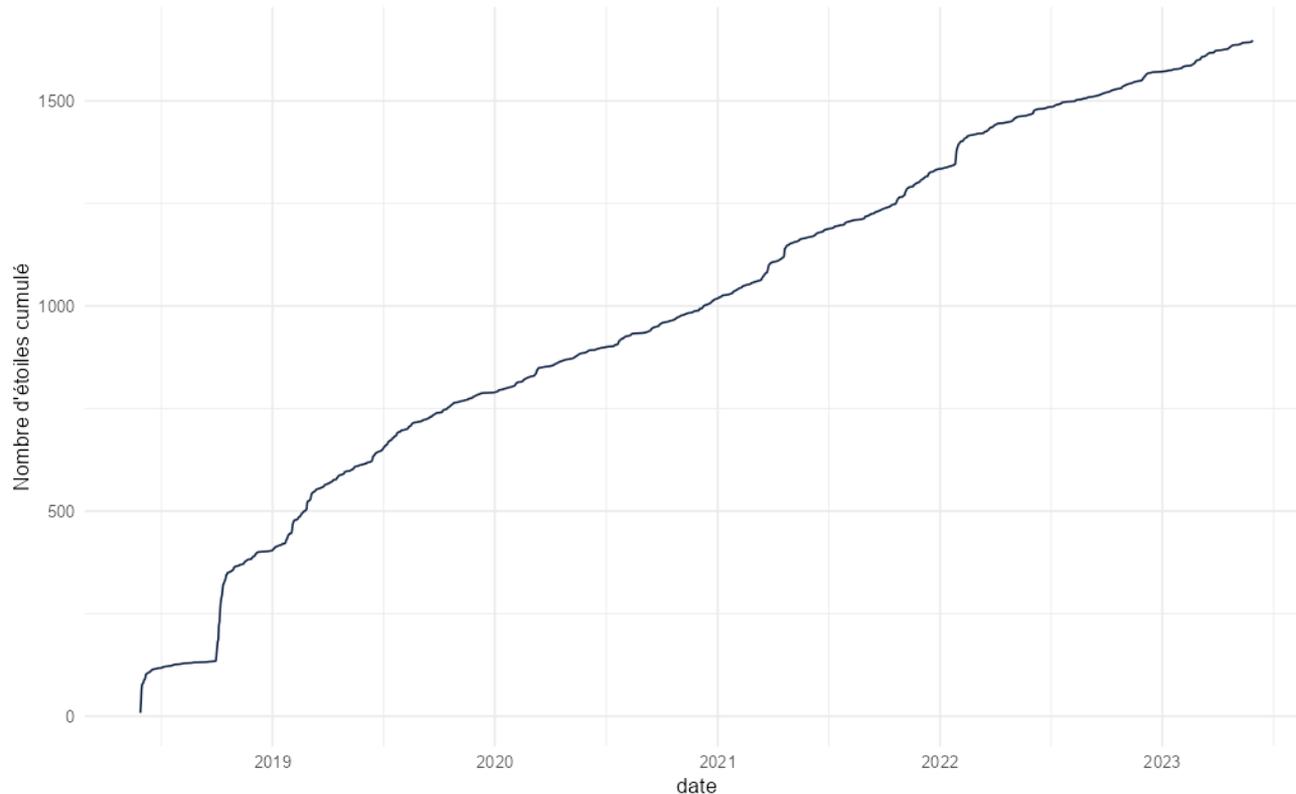
GitHub



Le code du paquet est disponible sur GitHub :

<https://github.com/dreamRs/esquisse>

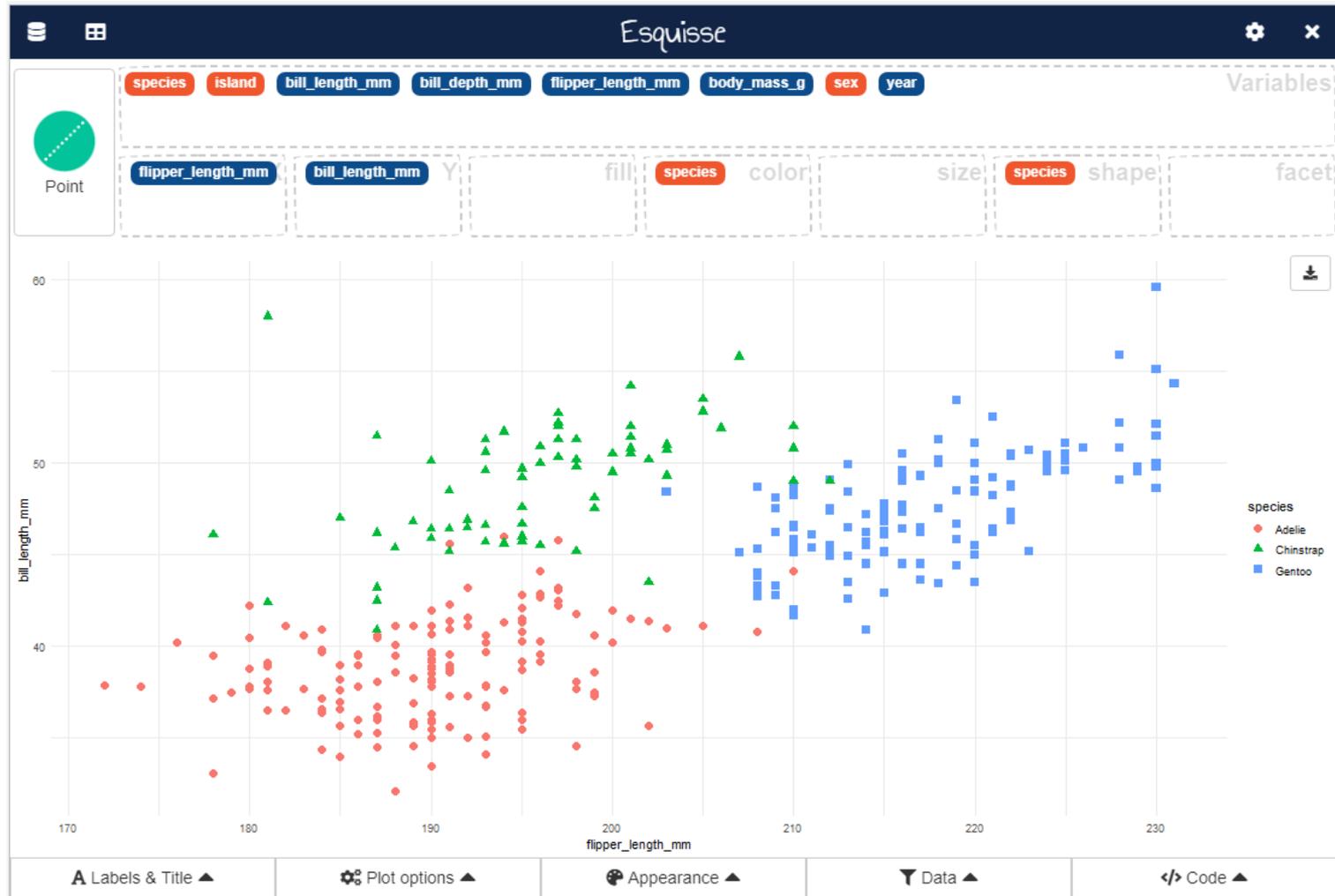
Nombre d'étoiles GitHub



Données : GitHub via thomas-neitmann/ghstars

esquisse

Interface graphique



esquisse

Comment l'utiliser ?



Sur le CRAN : <https://cran.r-project.org/package=esquisse>

Lancer l'application :

```
esquisse::esquisser()
```

En spécifiant un jeu de données :

```
esquisse::esquisser(my_data_frame)
```



esquisse



Principales fonctionnalités

Voir les données

Modifier les données

Sélection des zones de glisser / déposer

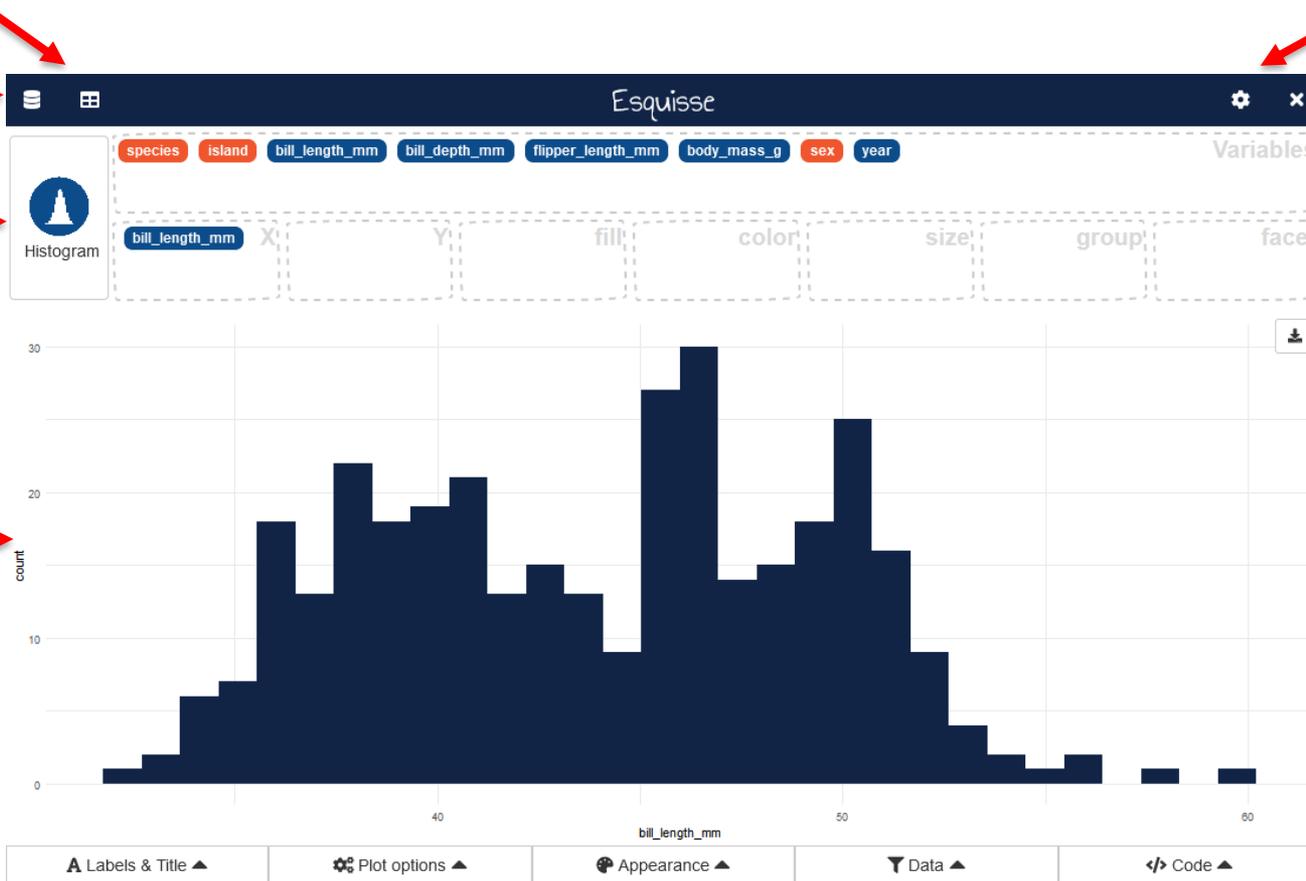
Type de graphique

Variables / noms de colonne

Variables à utiliser dans le graphique

Menu pour exporter le graphique

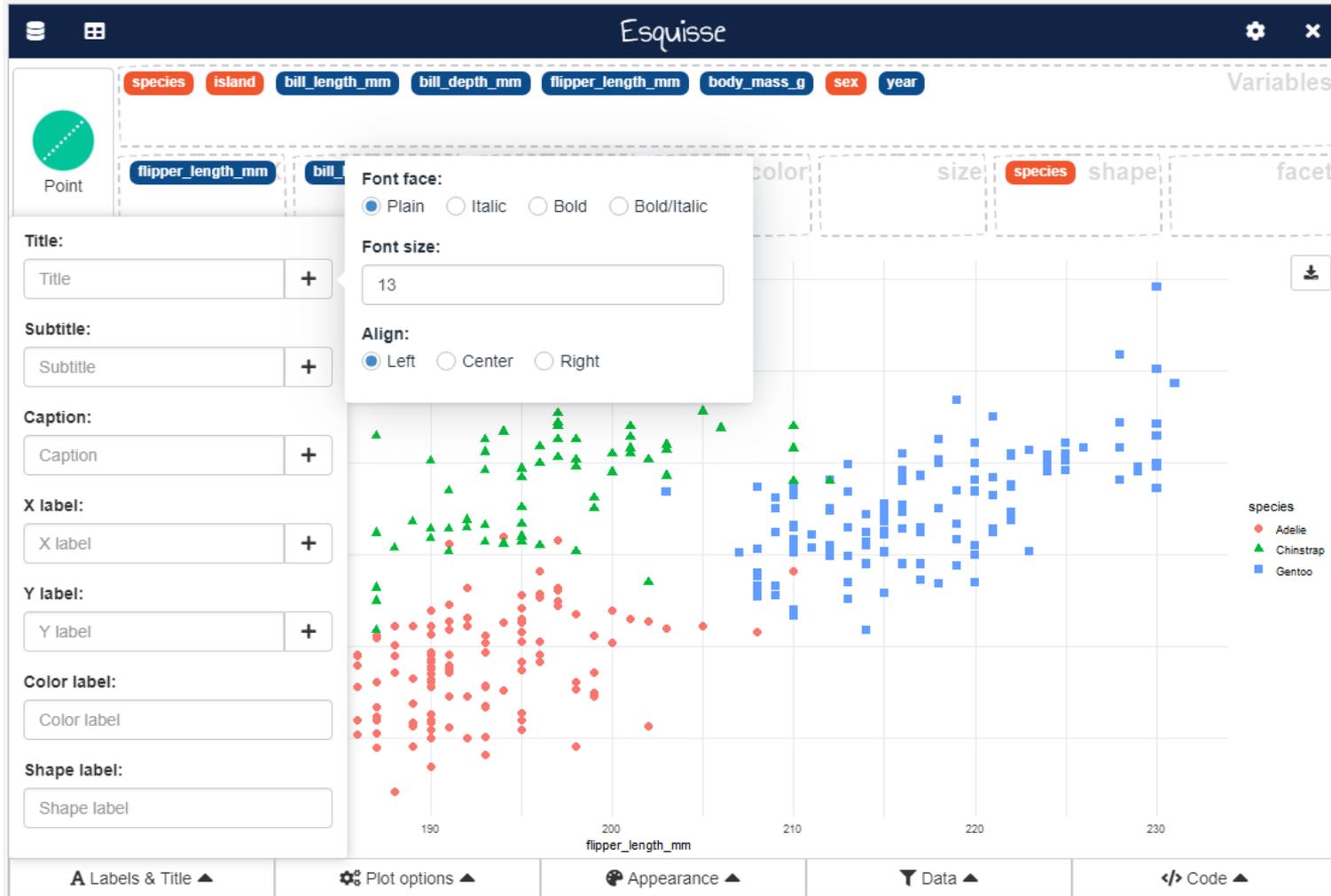
Graphique



Paramètres pour modifier le graphique et pour obtenir le code

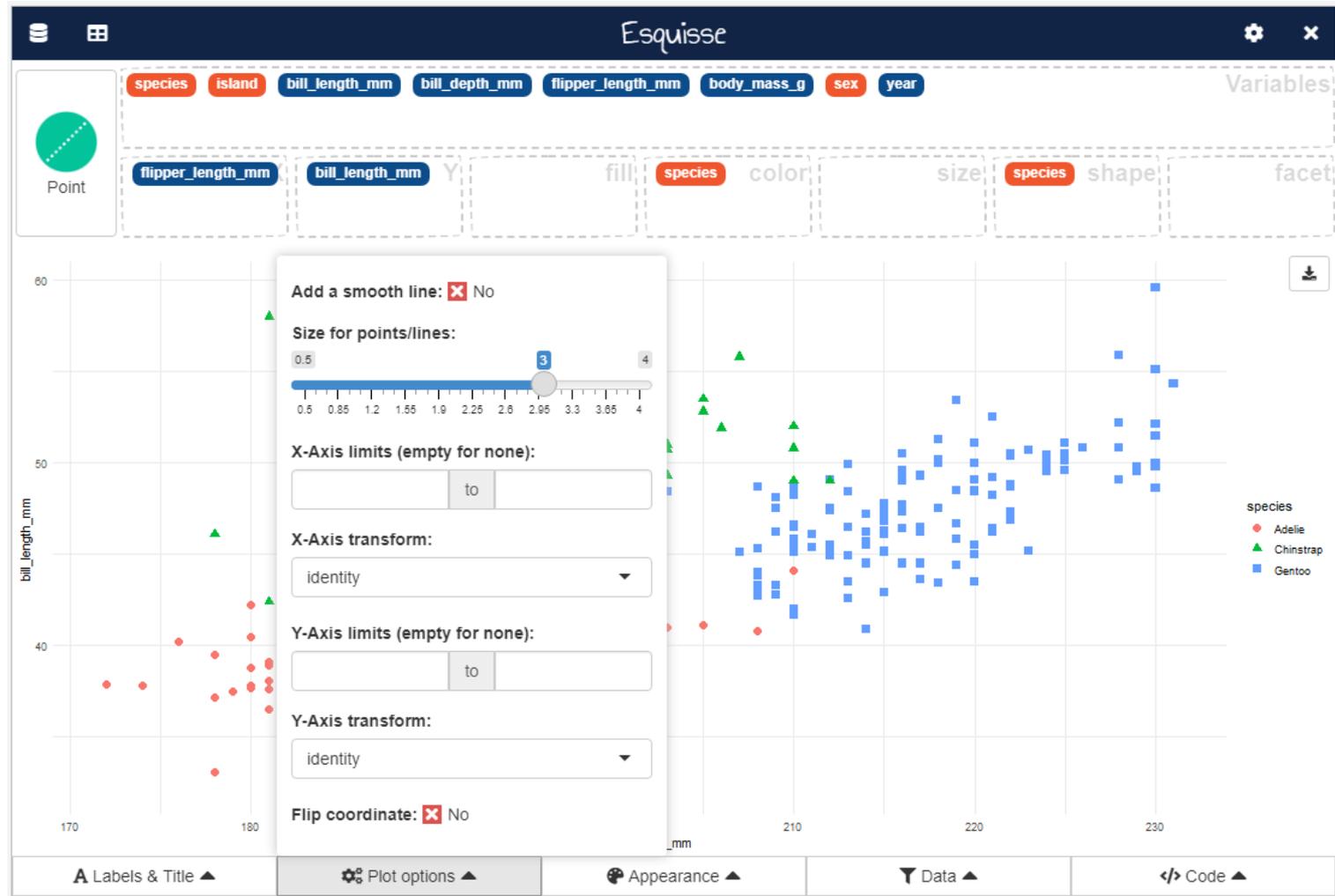
esquisse

Titre et libellés



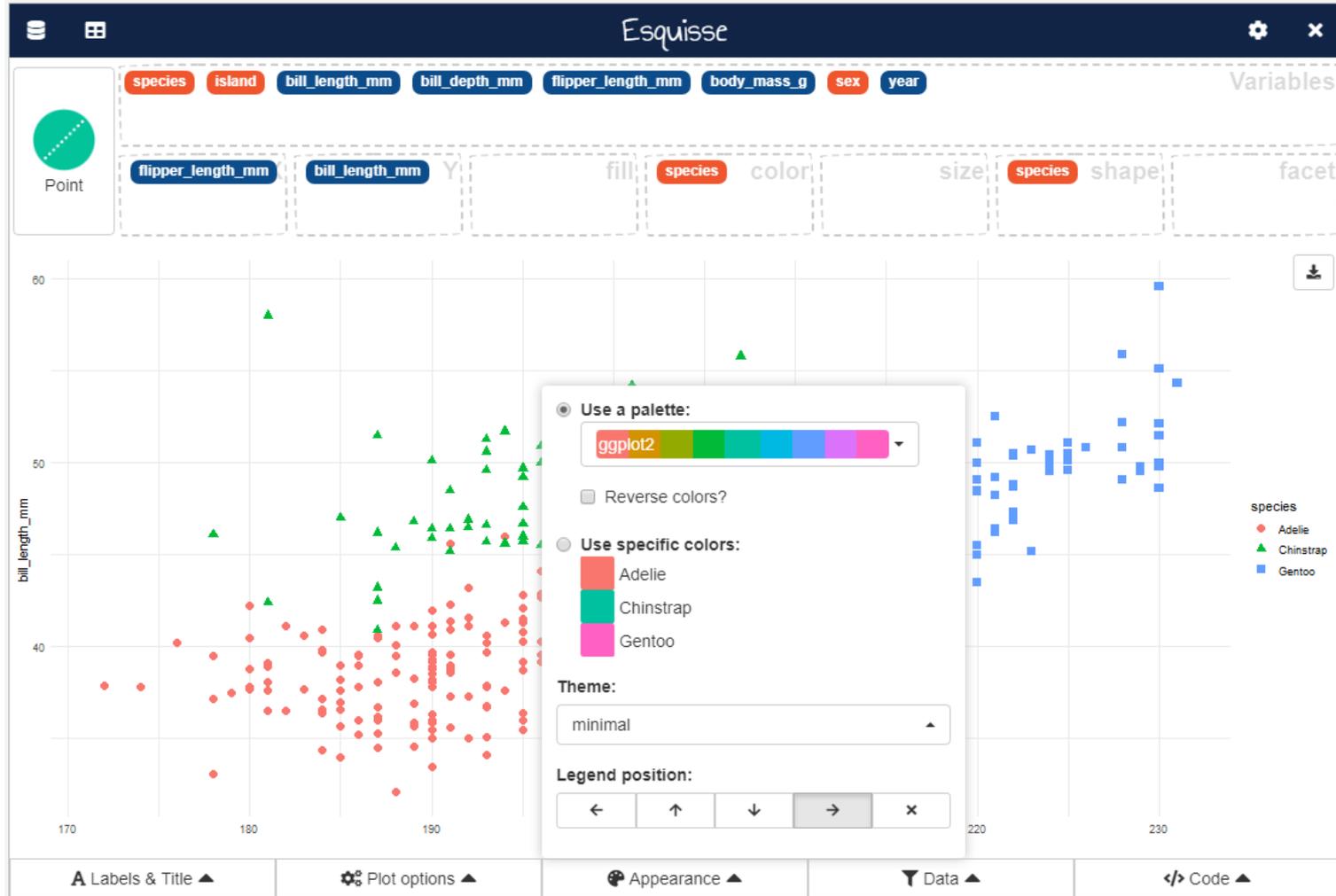
esquisse

Paramètres



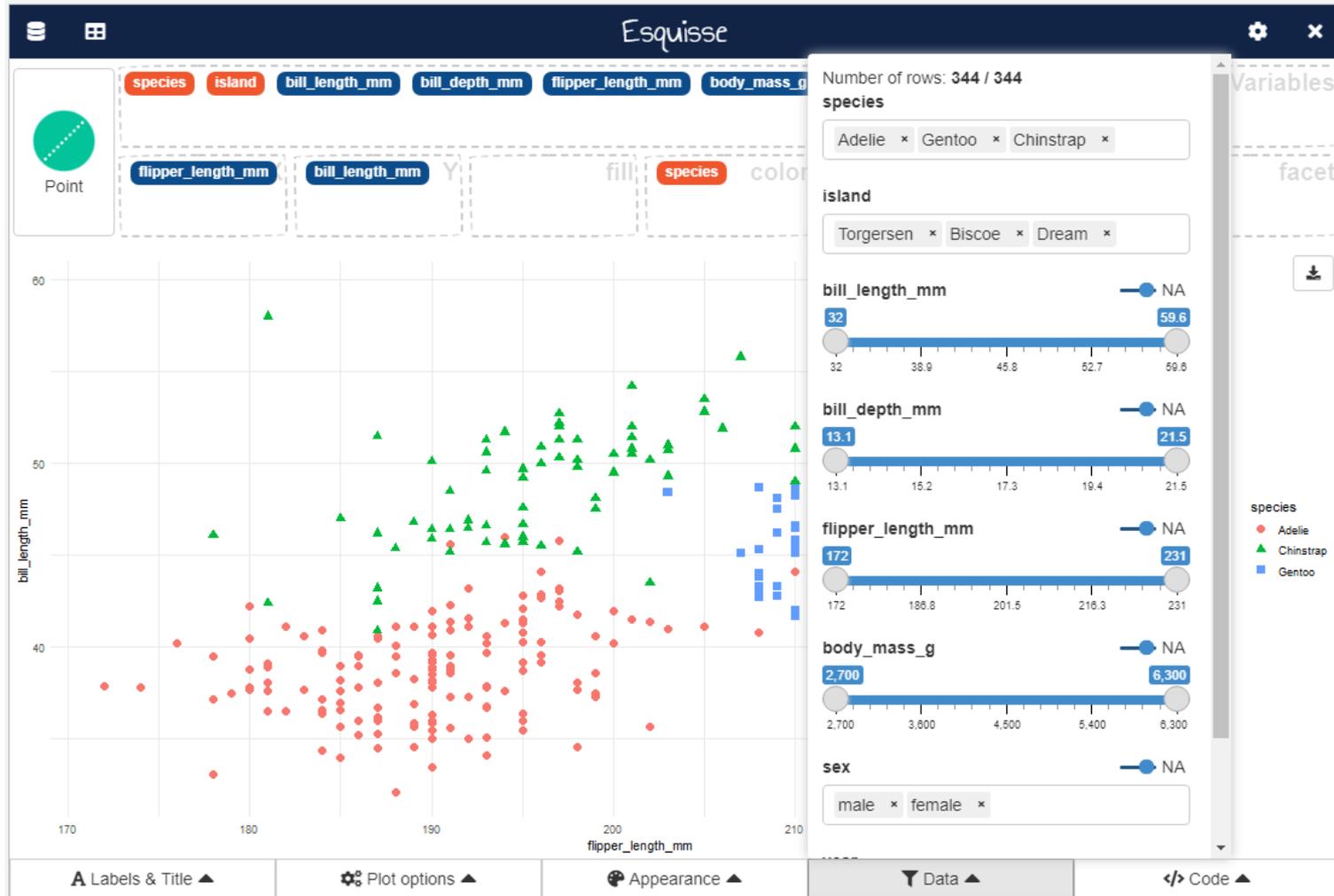
esquisse

Couleurs et thèmes



esquisse

Filtres



esquisse

Code



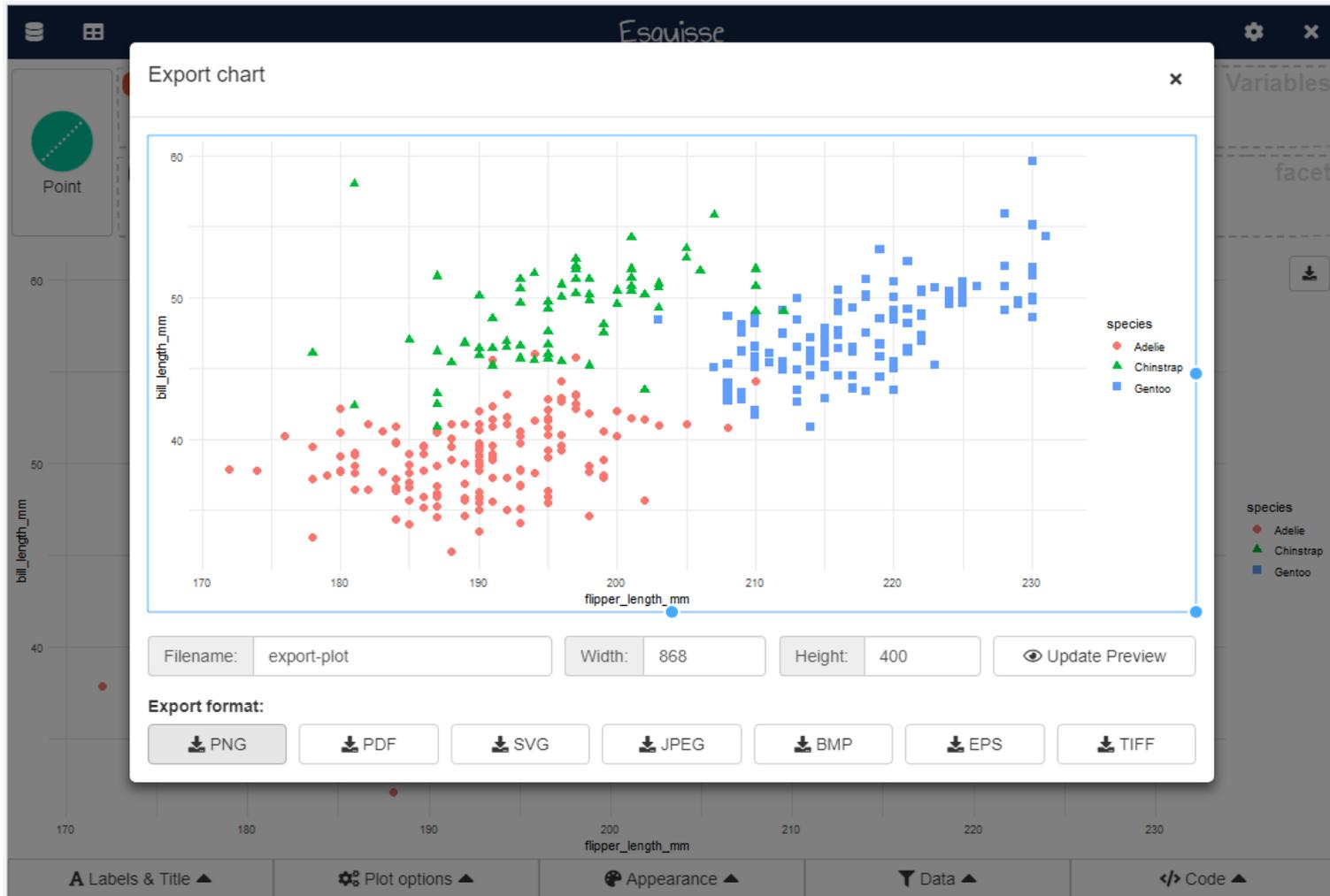
The screenshot shows the Esquisse software interface. At the top, the title bar reads "Esquisse". Below it is a "Variables" panel with buttons for "species", "island", "bill_length_mm", "bill_depth_mm", "flipper_length_mm", "body_mass_g", "sex", and "year". A "Point" legend is on the left. The main plot area is titled "Flipper and bill length" and displays a scatter plot of "bill_length_mm" (y-axis, 40-60) versus "flipper_length_mm" (x-axis, 170-210). The plot uses three species: Adelie (orange circles), Chinstrap (purple triangles), and Gentoo (teal squares). A "Code" panel on the right contains the following R code:

```
ggplot(`palmerpenguins::penguins`) +  
  aes(  
    x = flipper_length_mm,  
    y = bill_length_mm,  
    colour = species,  
    shape = species  
  ) +  
  geom_point(size = 3L) +  
  scale_color_manual(  
    values = list(  
      Adelie = "#FF9113",  
      Chinstrap = "#B455F3",  
      Gentoo = "#158F90"  
    )  
  ) +  
  labs(  
    title = "Flipper and bill length",  
    caption = "Data from {palmerpenguins} pack  
age",  
    color = "Penguin species",  
    shape = "Penguin species"  
  ) +  
  theme_minimal() +  
  theme(plot.title = element_text(size = 15L,  
    face = "bold"))
```

Below the code panel is a button "Insert code in script". At the bottom of the interface are five tabs: "Labels & Title", "Plot options", "Appearance", "Data", and "Code".

esquisse

Export



datamods

Import de données

Import data to be used in application

Import View Update

How to import data?

- Environment
- External file
- Copy / Paste
- Googlesheets

Upload a file:

Browse... pop-fra-reg-dep.xls

Upload complete

Select sheet to import:

Départements

✓ Data successfully imported! data has 100 below:

Code région <i>character</i>	Nom de la région <i>character</i>		
84	Auvergne-Rhône-Alpes		
32	Hauts-de-France	02	Aisne
84	Auvergne-Rhône-Alpes	03	Allier
93	Provence-Alpes-Côte d'Azur	04	Alpes-de-Haute-Provence
93	Provence-Alpes-Côte d'Azur	05	Hautes-Alpes

Number of rows to skip before reading data:
n = 6

Decimal separator:
0.00 .

Encoding:
A UTF-8

Import data

esquisse



**Des modules et encore des
modules (mais pas que)**

Modules {shiny}

Tous les composants constituant l'application sont sous forme de modules {shiny} :

```
name_ui <- function(id) {  
  ns <- NS(id)  
  tagList( )  
}  
name_server <- function(id) {  
  moduleServer(  
    id,  
    function(input, output, session) { }  
  )  
}
```



Modules réutilisables

Dans {esquisse} :

- Esquisse lui-même : `esquisse_ui` / `esquisse_server`
- Export de graphiques : `save_ggplot_ui` / `save_ggplot_server`
- Affichage d'un graphique avec export : `ggplot_output` / `render_ggplot`

Mais aussi des fonctions “input” utilisables dans une app {shiny} :

- Interface de glisser – déposer : `dragulaInput`
- Menu de selection de couleur ou de palette : `colorPicker` / `palettePicker`

Module shiny



Vous pouvez utiliser {esquisse} dans vos applications en utilisant le module :

```
# UI
esquisse_ui(
  id = "esquisse",
  header = FALSE,
  container = esquisseContainer(height = "700px")
)

# Server
esquisse_server(
  id = "esquisse",
  data_rv = reactiveValues(data = mtcars, name = "mtcars")
)
```



Menu pour exporter un graphique



Voir la page d'aide : ?` save-ggplot-module`

Save plot ✕

Filename: Width: Height:

Export format:

Interface de glisser - déposer



Voir la page d'aide : ?dragulaInput

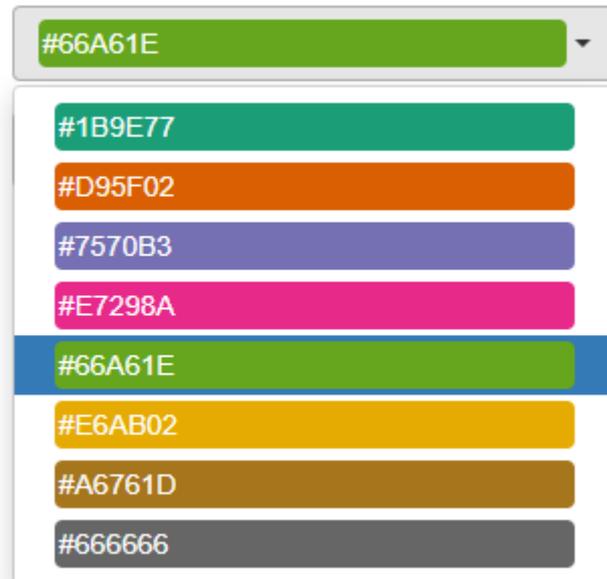
cyl disp hp wt qsec vs am gear carb			Source
mpg	x	y	color
fill	drat	size	facet

```
List of 2
$ source: chr [1:9] "cyl" "disp" "hp" "wt" ...
$ target:List of 6
..$ x : chr "mpg"
..$ y : NULL
..$ color: NULL
..$ fill : NULL
..$ size : chr "drat"
..$ facet: NULL
```

Sélection de couleurs / palettes



Choose a color:



`colorPicker()`

Choose a palette



`palettePicker()`

Le paquet {datamods}

{datamods} est le paquet compagnon d'esquisse, disponible sur le CRAN : <https://cran.r-project.org/package=datamods>

Les fonctionnalités suivantes sont disponibles sous forme de modules {shiny} :

- Import de données dans une application en utilisant :
 - L'environnement global ou celui d'un paquet
 - Un fichier texte, CSV, Excel, ...
 - En copiant / collant des données
 - Un fichier Google Sheet
- Filtres sur un data.frame
- Sélection et renommage de variables
- Édition de données (non intégré à esquisse)
- Échantillonnage de données (non intégré à esquisse)

Import de données



Voir la page d'aide : ?`import-modal`

Import data to be used in application ✕

↓ Import ☰ View ⚙ Update

How to import data?

- </> Environment
- External file
- Copy / Paste
- 📄 Google sheets
- 🔗 URL

Upload a file:

Browse... pop-fra-dep.txt ⚙

Upload complete

✓ **Data successfully imported!** data has 100 obs. of 9 variables. First five rows are shown below:

Code région <i>integer</i>	Nom de la région <i>character</i>	Code département <i>character</i>	Nom du département <i>character</i>
84	Auvergne-Rhône-Alpes	01	Ain
32	Hauts-de-France	02	Aisne
84	Auvergne-Rhône-Alpes	03	Allier
93	Provence-Alpes-Côte d'Azur	04	Alpes-de-Haute-Provence
93	Provence-Alpes-Côte d'Azur	05	Hautes-Alpes

< >

➔ Import data

Import de données



Voir la page d'aide : ?` import-modal`

Import data to be used in application ✕

[↓ Import](#) [View](#) [Update](#)

Code région <i>integer</i>	Nom de la région <i>character</i>	Code départemen t <i>character</i>	Nom du départemen t <i>character</i>	Nombre d'arrondiss ements <i>integer</i>	Nombre de cantons <i>integer</i>	Nombre de communes <i>integer</i>	Population municipale <i>character</i>	Populati on totale <i>character</i>
84	Auvergne-Rhône-Alpes	01	Ain	4	23	393	643 350	659 180
32	Hauts-de-France	02	Aisne	5	21	800	534 490	546 527
84	Auvergne-Rhône-Alpes	03	Allier	3	19	317	337 988	347 035
93	Provence-Alpes-Côte d'Azur	04	Alpes-de-Haute-Provence	4	15	198	163 915	168 381
93	Provence-Alpes-Côte d'Azur	05	Hautes-Alpes	2	15	162	141 284	145 883
93	Provence-Alpes-Côte d'Azur	06	Alpes-Maritimes	2	27	163	1 083 310	1 097 490
84	Auvergne-Rhône-Alpes	07	Ardèche	3	17	335	325 712	334 688
44	Grand Est	08	Ardennes	4	19	449	273 579	280 032
76	Occitanie	09	Ariège	3	13	327	153 153	157 210
44	Grand Est	10	Aube	3	17	431	310 020	317 118

< 1-10 of 100 rows Previous **1** 2 3 4 5 ... 10 Next

[↪ Import data](#)

Import de données



Voir la page d'aide : ?` import-modal`

Import data to be used in application ✕

[↓ Import](#) [☰ View](#) [⚙ Update](#)

Data has 100 observations and 9 variables. ⚙

<input checked="" type="checkbox"/>	Name	Class	New class	Missing values	Complete obs.	Unique values	^
<input checked="" type="checkbox"/>	Code région	integer	integer ▾	0	100,0 %	17	
<input checked="" type="checkbox"/>	Nom de la régic	character	character ▾	0	100,0 %	17	
<input checked="" type="checkbox"/>	Code départer	character	character ▾	0	100,0 %	100	
<input checked="" type="checkbox"/>	Nom du départi	character	character ▾	0	100,0 %	100	
<input checked="" type="checkbox"/>	Nombre d'aror	integer	integer ▾	0	100,0 %	7	
<input checked="" type="checkbox"/>	Nombre de can	integer	integer ▾	3	97,0 %	17	▾

ℹ Select, rename and convert variables in table above, then apply changes by clicking button below.

[↻ Apply changes](#)

[↻ Import data](#)

Filtrer des données



Voir la page d'aide : ?`filter-data`

Number of rows: 64 / 93

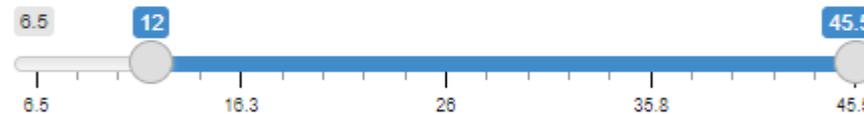
Manufacturer

Acura × Audi × BMW × Buick × Cadillac ×
Chevrolet × Chrysler × Chrysler × Dodge × Eagle ×
Ford × Geo × Honda × Hyundai × Infiniti ×

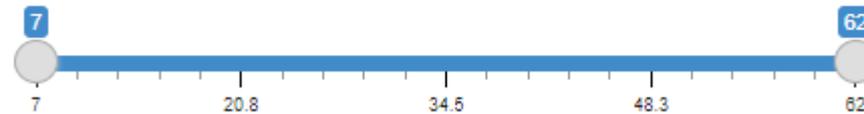
Type

Small × Midsize × Compact × Large × Sporty ×
Van ×

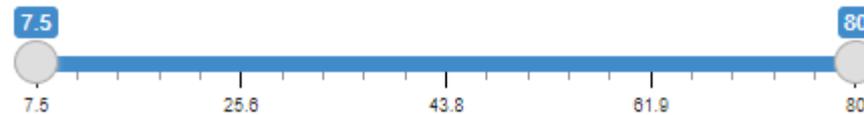
Min.Price



Price



Max.Price



Éditer des données



Voir la page d'aide : ?`edit-data`

Excel CSV

name	job	contactless_card	Update	Delete
Nery Stroman	Product designer	false		
Verla Cummerata-Lubowitz	Accommodation manager	false		
Francisquita Paucek	Sound technician, broadcasting/film /video	true		
Rosina Armstrong	Psychotherapist	true		
Marian West PhD	Theatre director	true		
Beulah Bailey	Curator	true		
Daron Denesik II	Designer, graphic	true		
Kristian Bahringer	Further education lecturer	true		
Pedro Feest	Hospital pharmacist	true		
Nanci Batz	Actor	false		

1-10 of 20 rows

'data.frame': 20 obs. of 6 variables:

Update row

name * :
Verla Cummerata-Lubowitz

job * :
Accommodation manager

credit_card_provider :
Mastercard

credit_card_security_code :
947

date_obtained :
January, 2018

SU	MO	TU	WE	TH	FR	SA
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3

contactless_card :

Validate the entry

+ Add a row

esquisse



Construction d'appels à des fonctions avec {rlang}

rlang

Générer du code

Au début : concatenation de chaînes de caractères

```
code <- function(data, mapping, geometry) {  
  data <- sprintf("ggplot(%s)", data)  
  mapping <- paste(  
    paste(names(mapping), unlist(mapping), sep = " = "),  
    collapse = ", "  
  )  
  mapping <- sprintf("aes(%s)", mapping)  
  geometry <- sprintf("geom_%s()", geometry)  
  paste(data, mapping, geometry, sep = " + ")  
}  
  
code("mpg", list(x = "displ", y = "hwy"), "point")  
#> [1] "ggplot(mpg) + aes(x = displ, y = hwy) + geom_point()"
```



rlang

Générer du code

Maintenant dans {esquisse}: construction d'un appel à une fonction avec {rlang}

```
library(rlang)
code <- function(data, mapping, geometry) {
  mapping <- expr(aes(!!!syms(mapping)))
  geometry <- call2(paste0("geom_", geometry))
  expr(ggplot(!!sym(data)) + !!mapping + !!geometry)
}

code("mpg", list(x = "displ", y = "hwy"), "point")
#> ggplot(mpg) + aes(x = displ, y = hwy) + geom_point()
```



La sortie est un “call” qu’il est possible d’évaluer.

Code généré dans {esquisse}

La fonction `ggcall()` est utilisée pour créer l'appel `ggplot`

```
library(esquisse)
library(ggplot2)
library(rlang)

# Création du call en fonction des paramètres de l'app
gg_c <- ggcall(
  data = "mtcars",
  mapping = list(x = "mpg", y = "wt"),
  geom = "point"
)

# Evaluation => produit le graphique
eval(gg_c)

# Code : on récupère le code en caractère
deparse(gg_c)

# Evaluation en changeant les données
eval_tidy(gg_c, list(mtcars = subset(mtcars, cyl == 4)))
```



esquisse

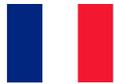


Internationalisation

Changer la langue



Aujourd'hui il est possible d'utiliser {esquisse} dans les langues suivantes :



- Français



- Allemand



- Chinois



- Espagnol



- Coréen



- Italien (*incomplet*)



- Albanais



- Macédonien



- Polonais



- Turc



- Portugais



- Ourdu (*incomplet*)

Changer la langue

On utilise la fonction `set_i18n()` :

```
library(esquisse)

# Choix de La Langue (code ISO 3166-2 du pays)
set_i18n("kr")

# Puis on lance esquisse comme d'habitude
esquisser()
```



Il faut déclarer la langue à utiliser avant de lancer l'application



Une évolution possible sera la possibilité de changer la langue directement dans l'interface

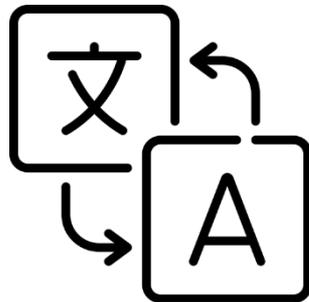
Changer la langue

Comment cela fonctionne ?

- Une fonction `i18n()` qui prend en argument un libellé à traduire
- Un dictionnaire (un par langue) avec la correspondance entre le libellé (en anglaise) et la traduction à utiliser

Exemple avec le fichier de traduction français :

<https://github.com/dreamRs/esquisse/blob/master/inst/i18n/fr.csv>



Contributions bienvenues !
(nouvelles traductions ou revue
de celles déjà présentes)

esquisse

Futurs développements



Interactivité



Session R dédiée

Possibilité de lancer
esquisse dans une
autre session R
(pour garder le
contrôle)

Des idées ?

Ouvrez un ticket
dans le dépôt
GitHub

esquisse



Les addins RStudio

WHAT?

Extension de RStudio



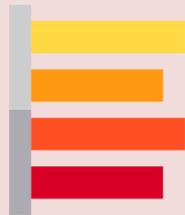
Faire des graphiques



Modifier ou créer des lignes de code



Modifier des objets



Créer des objets, des fichiers, des répertoires



Aussi loin que
votre
imagination
vous portera ...

WHY?

Accélérer des tâches répétitives
et

Simplifier des manipulations complexes

Fonctions pour interagir avec les scripts via {rstudioapi}

Ajout d'éléments, modification du code



Shiny gadgets via {miniUI}

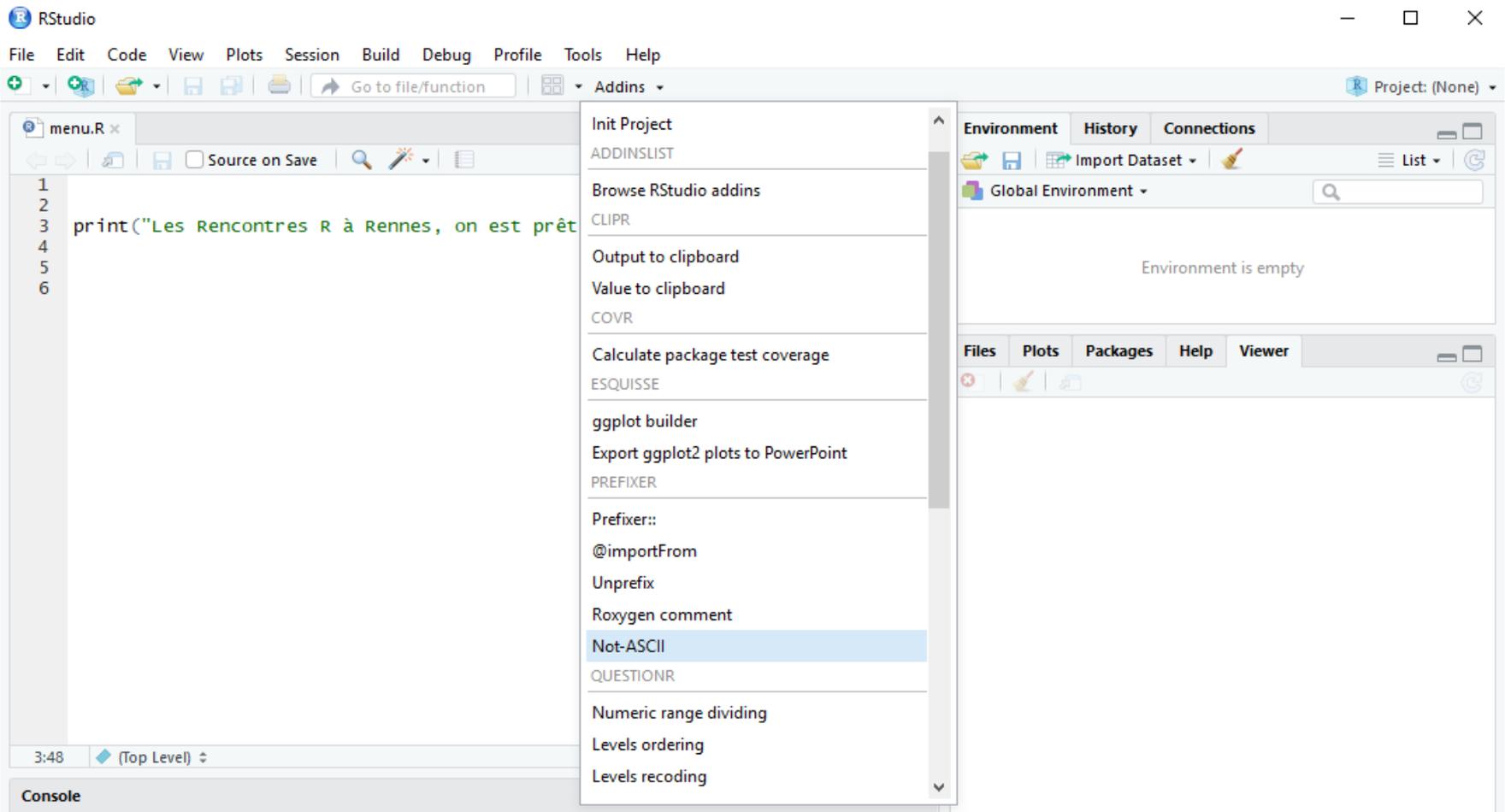
Interface pour interagir avec l'utilisateur



WHERE



Le menu des addins et les raccourcis claviers



The screenshot displays the RStudio interface with the 'Addins' menu open. The menu items are as follows:

- Init Project
- ADDINSLIST
- Browse RStudio addins
- CLIPR
 - Output to clipboard
 - Value to clipboard
- COVR
- Calculate package test coverage
- ESQUISSE
- ggplot builder
- Export ggplot2 plots to PowerPoint
- PREFIXER
 - Prefixer::
 - @importFrom
 - Unprefix
 - Roxygen comment
 - Not-ASCII**
 - QUESTIONR
- Numeric range dividing
- Levels ordering
- Levels recoding

The 'Not-ASCII' option is currently selected. The background shows the RStudio editor with a script named 'menu.R' containing the following code:

```
1  
2  
3 print("Les Rencontres R à Rennes, on est prêt  
4  
5  
6
```

The right-hand side of the interface shows the 'Environment' pane with 'Global Environment' and the message 'Environment is empty'. The bottom status bar indicates the time is 3:48 and the current level is '(Top Level)'.

WHO?

Tous les utilisateurs de Rstudio (développeurs, analystes, ...)



Contrairement aux applications Shiny souvent destinés aux utilisateurs finaux, les gadgets Shiny sont des outils pour interagir avec l'environnement Rstudio de l'utilisateur.

{addinit}



initialisation de projets RStudio normés.

Help **Init Project** Cancel

Create folders

Folders : scripts datas funs inputs outputs logs

Others :

Create scripts

Where :

Script's name :

By : Title :

Packages to load :

Config Script

{questionr}



Icut : découpage de variables continues

Cancel Done

Découpage interactif

Statistiques de `iris$Sepal.Length` :

	1st quartile	Median	Mean	3rd quartile	Max	NA
Min	4.3	5.1	5.8	5.8433	6.4	7.9
Max						0

Méthode
Quantile

Nombre d'intervalles
4

Breaks
4.3,5.1,5.8,6.4,7.9

Intervalles fermés à droite (`right`)
 Inclure la valeur extrême (`include.lowest`)
 Ajouter les valeurs extrêmes si nécessaire

Variable originale

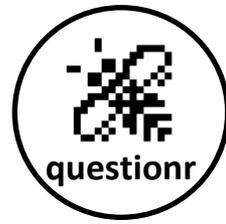
Frequency

iris\$Sepal.Length

Variable et paramètres Découpage en classes Code et résultat

Julien Barnier : <https://github.com/juba/questionr>

{questionr}



lorder : réordonnement des levels d'un factor

Cancel Interactive levels ordering Done

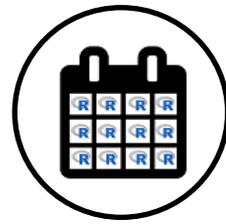
- Autre
- Ouvrier specialise
- Ouvrier qualifie
- Employe
- Technicien
- Profession intermediaire
- Cadre

Variable and settings Ordering Code and result

The image shows a screenshot of the 'Interactive levels ordering' interface. At the top, there is a header bar with 'Cancel' on the left, 'Interactive levels ordering' in the center, and 'Done' on the right. Below this is a large light gray area containing a list of seven job levels, each in a white box with a small cross icon on the left: 'Autre', 'Ouvrier specialise', 'Ouvrier qualifie', 'Employe', 'Technicien', 'Profession intermediaire', and 'Cadre'. At the bottom of the interface is a navigation bar with three icons: a list icon labeled 'Variable and settings', a cross icon labeled 'Ordering', and a code icon labeled 'Code and result'.

Julien Barnier : <https://github.com/juba/questionr>

{cronR}



Définir l'ordonnancement de scripts

Cancel Use cron to schedule your R script Done

Choose your Rscript

Select file

Selected Rscript
/home/jwiiffels/mycode.R

Job description
I get data from productxyz

Launch date:
2018-03-07

Job tags
hello world

Launch hour:
11:39

Additional arguments to Rscript
arg1 arg2 productxyz

Schedule:

- ONCE
- EVERY MINUTE
- EVERY HOUR
- EVERY DAY
- EVERY WEEK
- EVERY MONTH
- ASIS

Job identifier
job_63b543ab33e4494462c

**Rscript repository path:
launch & log location**
/home/bnosac

ASIS cron schedule
*/15 ****

Create job

Upload and create new jobs Manage existing jobs

{prefixer}



Ajout du namespace devant les fonctions et autres outils pour faire des packages.

The screenshot shows the RStudio IDE interface. The script editor on the left contains the following R code:

```
1  
2  
3 fread_dir <- function(path, pattern = "\\*.csv$") {  
4   paths <- list.files(path = path, pattern = pattern, full.names = TRUE)  
5   files <- lapply(paths, fread)  
6   files <- setNames(files, paths)  
7   rbindlist(l = files, idcol = "path")  
8 }  
9  
10
```

The dialog box on the right is titled "prefixer::" and has a "Cancel" button. It contains the following fields:

- Function:** A text box containing "fread".
- Context:** A text box containing "files <- lapply(paths, fread)".
- Suggested prefix:** A text box containing "data.table::fread".

At the bottom of the dialog, there are two buttons: "Add prefix" and "Skip". Below these buttons, there is a link "Skip all occurrences" and a note: "If your script use unloaded packages, functions from these packages won't be prefixed. You can load packages".

dreamRs : <https://github.com/dreamRs/prefixer>

HOW?

2 types d'addin

Fonctions pour interagir avec les scripts via `{rstudioapi}`

Exécution d'une fonction interagissant avec le script courant ou exécution de tâches prédéfinies

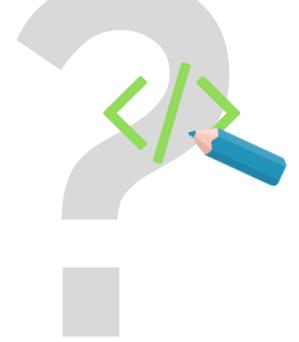
Shiny gadgets via `{miniUI}` (et `{shiny}` et `{leaflet}`)

Applications `{shiny}` permettant à l'utilisateur d'interagir avec une interface

Liste d'addins disponibles : <https://github.com/daattali/addinslist>

HOW

Développer un addin



- 1 Écrire une fonction ou développer une application
- 2 Créer un package contenant la fonction R
- 3 Enregistrer l'addin dans RStudio

pour cela il faut créer un fichier « .dcf » :

- Le fichier doit être situé dans `inst/rstudio/addins.dcf` (et doit s'appeler `addins.dcf`)
- Il doit contenir les informations suivantes :

```
DESCRIPTION x  addins.dcf x  README.md x  reorder_library.R x  geocode_addin.R x
1 Name: Réorganise library()
2 Description: Positionne les appels à library() en début de script.
3 Binding: reorder_library
4 Interactive: false
5
```

Merci



Contact

Twitter : [@dreamrs_fr](https://twitter.com/dreamrs_fr)

Galerie Shiny : <http://shinyapps.dreamrs.fr/>

Site web : <https://www.dreamrs.fr/>

GitHub : <https://github.com/dreamRs>