

Inférence statistique avec R: un cas d'étude

Jean-Philippe Morissette

Assistant de recherche

Groupe de recherche interdisciplinaire en
informatique de la santé – Uds

Statisticien

Chargé de cours

Département de mathématique, Uds
École de gestion, Uds



Table des matières

- Introduction
- Initiation à R
- Exploration d'un jeu de données: *Données sur le diabète gestationnel*
- Conclusion



Introduction

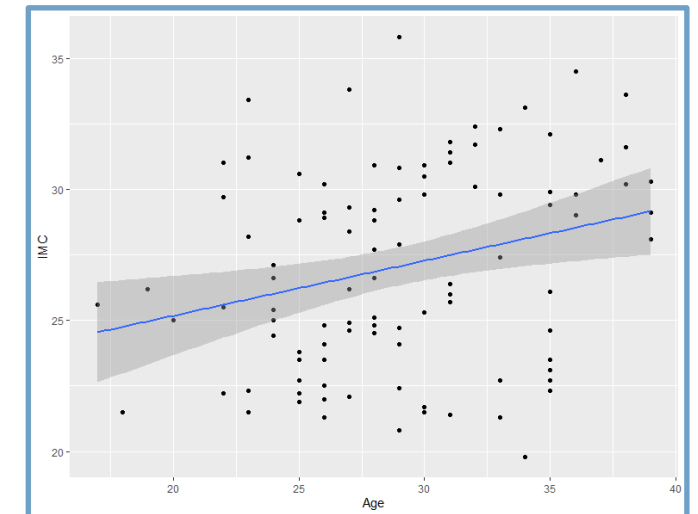
Qu'est-ce que R?

Characteristic	N = 102 [†]
Traitement	
Controle	51 / 102 (50%)
Intervention	51 / 102 (50%)
Gluc_T1	
N Non-missing/No. obs. (% Non-missing)	102.0/102.0 (100.0)
Mean (SD)	7.6 (0.8)
Median (Q1, Q3)	7.7 (7.1, 8.2)
Min, Max	5.9, 9.4
Gluc_T2	
N Non-missing/No. obs. (% Non-missing)	102.0/102.0 (100.0)
Mean (SD)	8.0 (0.8)
Median (Q1, Q3)	8.0 (7.5, 8.5)
Min, Max	6.2, 10.1



Characteristic	Beta	95% CI	p-value
Age	0.01	-0.01, 0.03	0.2
IMC	-0.01	-0.03, 0.01	0.4
Predia			
Non	—	—	
Oui	-0.04	-0.22, 0.14	0.6
Groupe_a_risque	0.32	0.06, 0.58	0.017
Abbreviation: CI = Confidence Interval			

```
1 classe ← classe %>%  
2 filter(BonsEtudiants==TRUE)
```



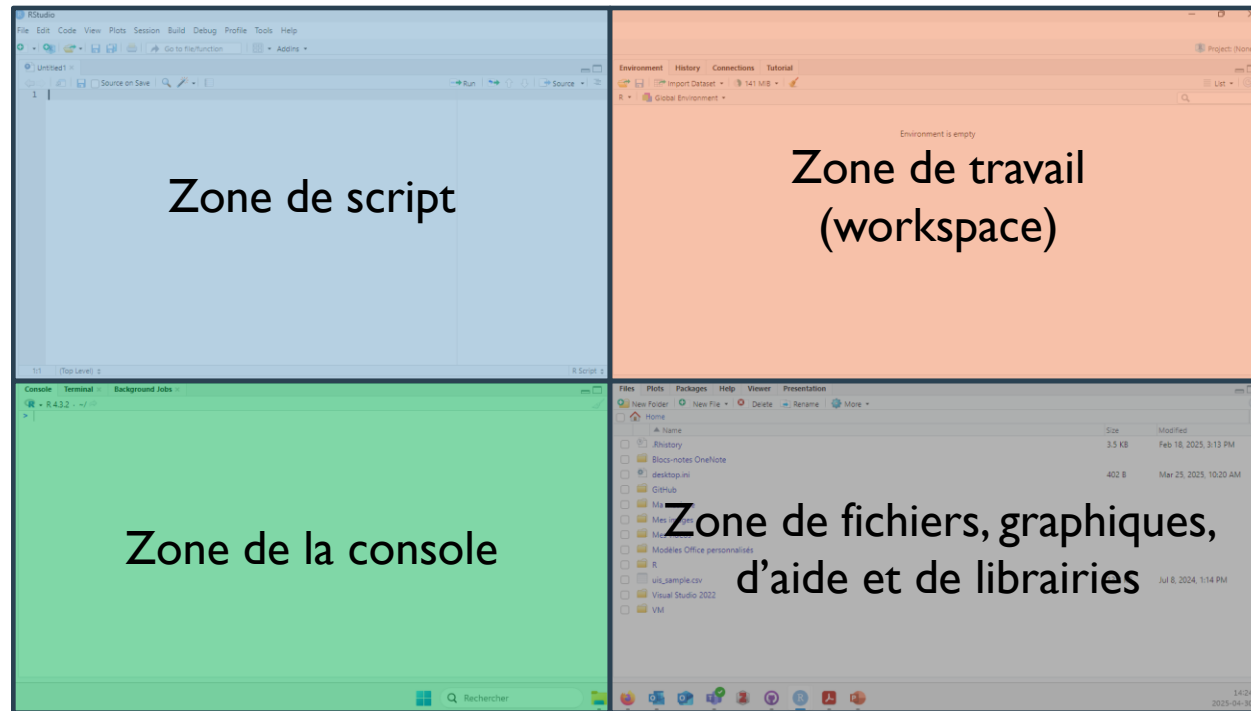
Pourquoi choisir R? (et pas python?)





Initiation à R

Utilisation de RStudio



Utilisation de RStudio... en trichant!

RStudio IDE : : CHEATSHEET

Documents and Apps

- Open Shiny, R Markdown, knitr, Sweave, LaTeX, Rd files and more in Source Pane
- Check spelling, output format options
- Render output
- Choose code chunk options
- Configure chunk options
- Insert code chunk
- Publish chunk to server
- Jump to previous chunk
- Jump to next chunk
- Run all previous code chunks
- Run this code chunk
- Set knitr chunk options
- Access markdown guide at **Help > Markdown Quick Reference**
- See reverse side for more on **Visual Editor**
- RStudio recognizes that files named **app.R**, **server.R**, **ui.R**, and **global.R** belong to a shiny app
- Run app
- Choose app location to shinyapps.io, publish
- View app
- Manage app on server
- accounts

Source Editor

- Navigate backwards/forwards
- Open in new window
- Save
- Find and replace
- Compile as notebook
- Run selected code
- Import data with wizard
- History of past commands to run/copy
- Manage external databases
- View memory usage
- R tutorials
- Multiple cursors/column selection with **Alt + mouse drag**
- Source with previous code
- Show file output pane or outline as a local job
- Code diagnostics that appear in the margin
- Hover over diagnostic symbols for details
- Syntax highlighting based on your file's extension
- Tab completion to finish function names, file paths, arguments, and more
- Multi-language code snippets to quickly use common blocks of code
- Jump to function in file
- Change file type
- Run scripts in separate sessions
- Maximize, minimize panes
- Ctrl/Cmd + arrow up** to see history
- Build Log
- Drag pane boundaries

Tab Panes

- Import data
- History of past commands to run/copy
- Manage external databases
- View memory usage
- R tutorials
- Multiple cursors/column selection with **Alt + mouse drag**
- Source with previous code
- Show file output pane or outline as a local job
- Code diagnostics that appear in the margin
- Hover over diagnostic symbols for details
- Syntax highlighting based on your file's extension
- Tab completion to finish function names, file paths, arguments, and more
- Multi-language code snippets to quickly use common blocks of code
- Jump to function in file
- Change file type
- Run scripts in separate sessions
- Maximize, minimize panes
- Ctrl/Cmd + arrow up** to see history
- Build Log
- Drag pane boundaries

Version Control

- Turn on at **Tools > Project Options > Git/SVN**
- Added
- Deleted
- Modified
- Renamed
- Untracked
- Stage files
- Commit
- Push/Pull
- View
- Open
- Close
- History
- Branch
- Open shell to type commands
- Show file diff to view file differences

Debug Mode

- Use **debug()**, **browser()**, or a breakpoint and execute your code to open the debugger mode
- Launch debugger mode from origin
- Open traceback to examine the functions that called before the error occurred
- Console
- Terminal
- Jobs
- Next
- Continue
- Stop
- Step through code one line at a time
- Step into and out of functions to run
- Resume execution
- Quit debug mode

Package Development

- Create a new package with **File > New Project > New Directory > R Package**
- Enable roxygen documentation with **Tools > Project Options > Build Tools**
- Roxygen guide at **Help > Roxygen Quick Reference**
- See package information in the **Build Tab**
- Install package and restart R
- Run devtools::load_all() and reload changes
- Run R CMD check
- Customize package build options
- Build from source
- Build binary package
- Run package tests
- Configure Build Tools

Plots

- RStudio opens plots in a dedicated **Plots** pane
- Navigate recent plots
- Open in recent window
- Export plot
- Delete plot
- Delete all plots

Help

- RStudio opens documentation in a dedicated **Help** pane
- Home page of helpful links
- Search within help file
- Search for help file

Viewer

- Viewer pane displays HTML content, such as Shiny apps, R Markdown reports, and interactive visualizations
- Stop Shiny app
- Publish to shinyapps.io, Post Connect, Post Cloud,...
- Refresh

View (data)

- View (data) opens spreadsheet-like view of data set
- Filter rows by value or value range
- Sort by values
- Search for value

Console

- Run commands in environment where execution has paused
- Examine variables in executing environment
- Select function in traceback to debug

Terminal

- Run commands in environment where execution has paused
- Examine variables in executing environment
- Select function in traceback to debug

Jobs

- Run commands in environment where execution has paused
- Examine variables in executing environment
- Select function in traceback to debug

Console

- Run commands in environment where execution has paused
- Examine variables in executing environment
- Select function in traceback to debug

Terminal

- Run commands in environment where execution has paused
- Examine variables in executing environment
- Select function in traceback to debug

Jobs

- Run commands in environment where execution has paused
- Examine variables in executing environment
- Select function in traceback to debug

Les feuilles de triches sont disponibles dans le répertoire de l'atelier!

À nous de joueR! (0)

Prérequis:

- *On suppose que vous avez R et RStudio d'installés sur votre machine.*
- *Vous avez téléchargé le répertoire de l'atelier sur votre machine.*

Si ce n'est pas déjà fait, ouvrez le fichier *AtelierR.R*. Si l'on vous demande avec quel logiciel vous souhaitez ouvrir le fichier, choisissez RStudio.

Nous allons explorer le tout ensemble pour commencer!

Quelques particularités à propos de R

- Assignment avec la flèche « <- »
- L'opérateur « c » pour la création de vecteurs
- Les commandes d'aide « ? » et « ?? »
- L'utilisation de librairies
- La lecture de fichiers au format .csv
- La structure *dataframe* et *tibble*

À vous de jouer! (1)

Que font les commandes suivantes?

- `head(df_glucose)`
- `tail(df_glucose)`
- `glimpse(df_glucose)`
- `view(df_glucose)`
- `df_glucose[1,]`
- `df_glucose[c(1,3,5),]`
- `df_glucose[,1]`
- `df_glucose[2,2]`
- `df_glucose[df_glucose$Predia==1,]`
- `df_glucose$Age`

Retour

Que font les commandes suivantes:

- `head(df_glucose)`
- `tail(df_glucose)`
- `glimpse(df_glucose)`
- `view(df_glucose)`
- `df_glucose[1,]`
- `df_glucose[c(1,3,5),]`
- `df_glucose[,1]`
- `df_glucose[2,2]`
- `df_glucose[df_glucose$Predia==1,]`
- `df_glucose$Age`

Affiche les premières lignes du jeu

Affiche les dernières lignes du jeu

Transposition de « head »

Ouvre le jeu de données dans R

Accède la ligne 1 du jeu

Accède les lignes 1, 3 et 5 du jeu

Accède la colonne 1 du jeu

Accède la ligne 2 et la colonne 2 du jeu

Accède les lignes où « Predia==1 »

Accède la colonne nommée « Age »

Le standard des données bien rangées

df	Variable 1	...	Variable j	...	Variable p
Individu 1					
...					
Individu i			df [i, j]		
...					
Individu n					

df [i,]

df [, j]



Manipulation de données

La majorité des manipulations de données que l'on doit faire rentrent dans l'une des catégories suivantes:

- Sélectionner des observations (donc des *lignes*!)
- Sélectionner des variables (donc des *colonnes*!)
- Ordonner les lignes
- Créer de nouvelles variables
- Créer des résultats sommaires (moyenne, proportion, nombre de cas, etc.)
- Créer des groupes

df	Variable l	...	Variable j	...	Variable p	
Individu l						
...						
Individu i			df [i, j]			df [i, j]
...						
Individu n			df [i, j]			

Manipulation de données

Action

Sélectionner des observations

Sélectionner des variables

Ordonner les lignes

Créer de nouvelles variables

Créer des résultats sommaires

Créer des groupes

Verbe

`filter(df, critère)`

`select(df, critère)`

`arrange(df, critère)`

`mutate(df, critère)`

`summarise(df, critère)`

`group_by(df, critère)`

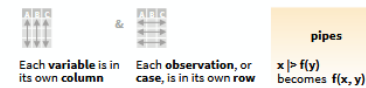


Manipulation de données... en trichant!

Data transformation with dplyr : : CHEATSHEET

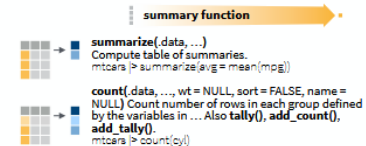


dplyr functions work with pipes and expect tidy data. In tidy data:



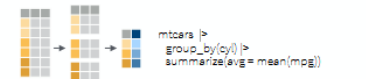
Summarize Cases

Apply **summary functions** to columns to create a new table of summary statistics. Summary functions take vectors as input and return one value (see back).

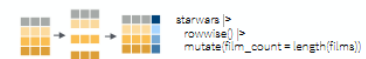


Group Cases

Use **group_by(data, ...)** add = FALSE, drop = TRUE to create a "grouped" copy of a table grouped by columns in ... dplyr functions will manipulate each "group" separately and combine the results.



Use **rowwise(data, ...)** to group data into individual rows. dplyr functions will compute results for each row. Also apply functions to list-columns. See tidy cheat sheet for list-column workflow.

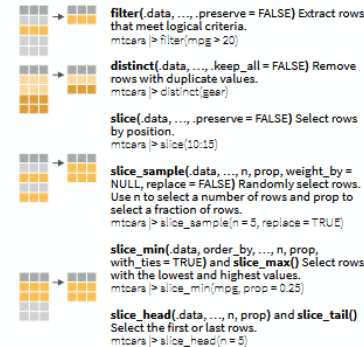


ungroup(k, ...) Returns ungrouped copy of table.
g_mtcars <- mtcars %> group_by(cyl)
ungroup(g_mtcars)

Manipulate Cases

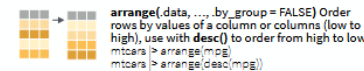
EXTRACT CASES

Row functions return a subset of rows as a new table.

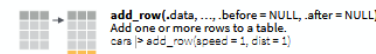


Logical and boolean operators to use with filter()
== < <= is.na() %in% | xor()
!= > >= !is.na() ! &
See ?base::Logic and ?Comparison for help.

ARRANGE CASES



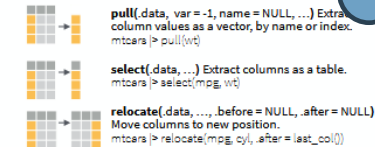
ADD CASES



Manipulate Variables

EXTRACT VARIABLES

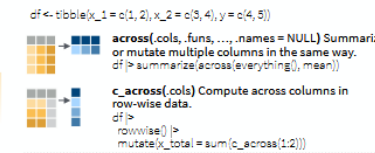
Column functions return a set of columns as a new vector.



Use these helpers with **select()** and **across()**

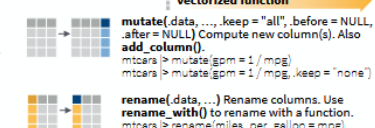
e.g. mtcars %> select(mpg:cyl)
contains(match) num_range(prefix, range) ; e.g. mpg:cyl
ends_with(match) all_of(x)/any_of(x, ..., vars) ; e.g. gear
starts_with(match) matches(match) ; e.g. everything()

MANIPULATE MULTIPLE VARIABLES AT ONCE



MAKE NEW VARIABLES

Apply **vectorized functions** to columns. Vectorized functions take vectors as input and return vectors of the same length as output (see back).



Les feuilles de triches sont disponibles dans le répertoire de l'atelier!

Manipulation de données (Exemples)

Exemples:

- Ordonnez le jeu de données en fonction de l'IMC des patientes.
- Créez une nouvelle variable qui correspond à la cote z des patientes pour ce qui est de leur IMC.
- Pour chaque catégorie de la variable traitement, calculez la moyenne de l'IMC des patientes.

Manipulation de données (Exemples)

Exemples:

- Ordonnez le jeu de données (*action*) en fonction de l'IMC des patientes (*critère*).
- Créez une nouvelle variable qui correspond à la cote z des patientes pour ce qui est de leur IMC.
- Pour chaque catégorie de la variable traitement, calculez la moyenne de l'IMC des patientes.

Manipulation de données (Exemples)

Exemples:

- Ordonnez le jeu de données (*action*) en fonction de l'IMC des patientes (*critère*).
- Créez une nouvelle variable (*action*) qui correspond à la cote z des patientes pour ce qui est de leur IMC (*critère*).
- Pour chaque catégorie de la variable traitement, calculez la moyenne de l'IMC des patientes.

Manipulation de données (Exemples)

Exemples:

- Ordonnez le jeu de données (*action*) en fonction de l'IMC des patientes (*critère*).
- Créez une nouvelle variable (*action*) qui correspond à la cote z des patientes pour ce qui est de leur IMC (*critère*).
- Pour chaque catégorie (*action*) de la variable traitement (*critère*), calculez (*action*) la moyenne de l'IMC des patientes (*critère*).

L'opérateur *pipe*

Le langage R dispose d'un opérateur particulier, appelé *pipe* « %>% ». Cet opérateur permet d'utiliser un objet comme **premier argument** d'une fonction, et peut être utilisé à répétition.

Par exemple,

```
df_glucose ← arrange(df_glucose, IMC)
```

est équivalent à

```
df_glucose ← df_glucose %>%  
  arrange(IMC).
```



Illustration de l'opérateur *pipe*

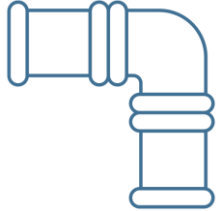
```
df_glucose %>%  
  arrange(IMC) %>%  
  filter(Predia==1)
```


L'opérateur *pipe* permet d'utiliser un objet comme **premier argument** d'une fonction, et peut être utilisé à répétition.

Illustration de l'opérateur *pipe*

```
df_glucose %>%  
  arrange(IMC) %>%  
  filter(Predia==1)
```

df_glucose

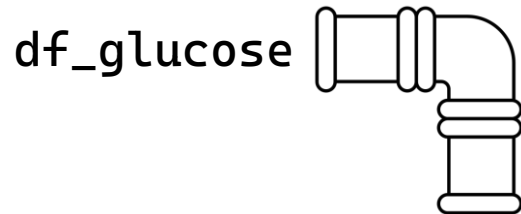


arrange(.data = , by_group = IMC)

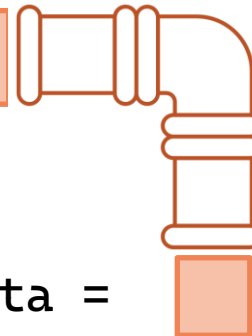
L'opérateur *pipe* permet d'utiliser un objet comme **premier argument** d'une fonction, et peut être utilisé à répétition.

Illustration de l'opérateur *pipe*

```
df_glucose %>%  
  arrange(IMC) %>%  
  filter(Predia==1)
```



```
arrange(.data = , by_group = IMC)
```

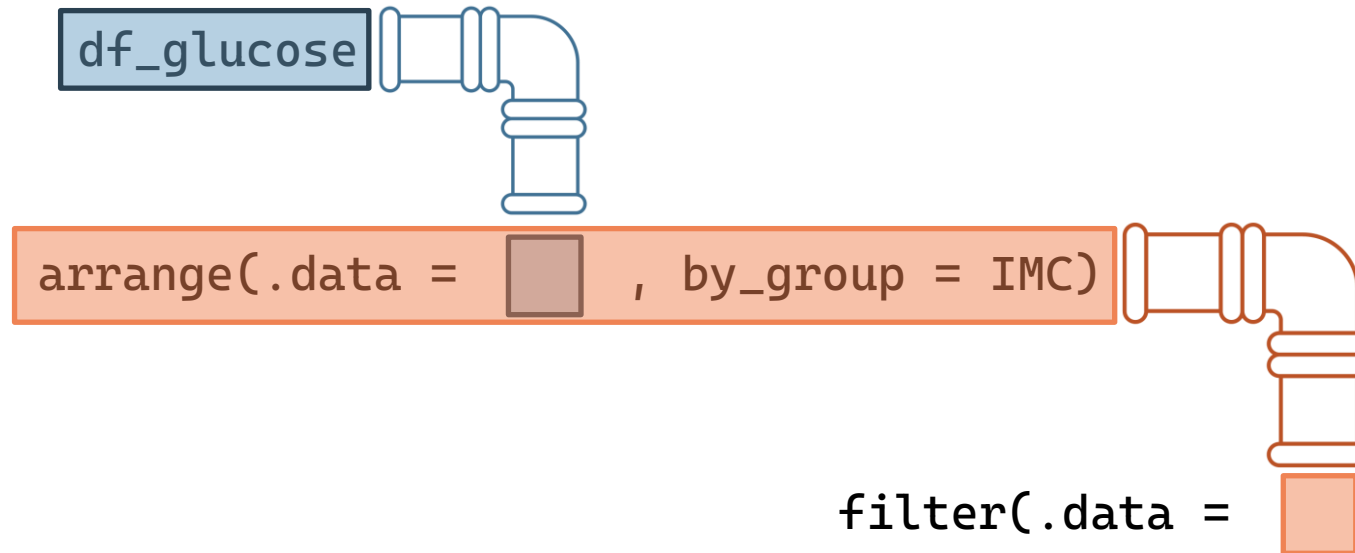


```
filter(.data = , .by = Predia==1)
```

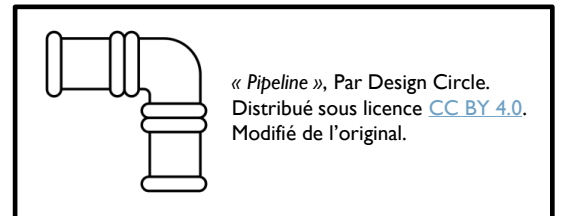
L'opérateur *pipe* permet d'utiliser un objet comme **premier argument** d'une fonction, et peut être utilisé à répétition.

Illustration de l'opérateur *pipe*

```
df_glucose %>%  
  arrange(IMC) %>%  
  filter(Predia==1)
```



L'opérateur *pipe* permet d'utiliser un objet comme **premier argument** d'une fonction, et peut être utilisé à répétition.



L'opérateur *pipe* (Exemple)

Énoncé initial:

Pour chaque catégorie de la variable traitement, calculez la moyenne de l'IMC des patientes.

L'opérateur *pipe* (Exemple)

Énoncé initial:

Pour chaque catégorie de la variable traitement (*créer des groupes*), calculez la moyenne de l'IMC des patientes (*créer des résultats sommaires*).

L'opérateur *pipe* (Exemple)

Énoncé initial:

Pour chaque catégorie de la variable traitement (*créer des groupes*), calculez la moyenne de l'IMC des patientes (*créer des résultats sommaires*).

Traduction en langage R:

```
df_glucose %>%  
  group_by(Traitement) %>%  
  summarise(mean(IMC))
```

L'opérateur *pipe* (Exemple)

Énoncé initial:

Pour chaque catégorie de la variable traitement (*créer des groupes*), calculez la moyenne de l'IMC des patientes (*créer des résultats sommaires*).

Traduction en langage R:

```
df_glucose %>%  
  group_by(Traitement) %>%  
  summarise(mean(IMC))
```

Traduction en langage usuel:

« À partir du jeu de données df_glucose, regrouper les observations en fonction du traitement, PUIS calculer la moyenne de l'IMC. »

Manipulation de données

Exemple:

Supposons que l'on souhaite calculer l'IMC moyen des patientes ayant entre 25 et 30 ans, et ce en comparant les patientes qui ont reçu le traitement à celles qui ne l'ont pas reçu.

Action

Sélectionner des observations
Sélectionner des variables
Ordonner les lignes
Créer de nouvelles variables
Créer des résultats sommaires
Créer des groupes

Verbe

`filter(df, critère)`
`select(df, critère)`
`arrange(df, critère)`
`mutate(df, critère)`
`summarise(df, critère)`
`group_by(df, critère)`

Manipulation de données

Exemple:

Supposons que l'on souhaite calculer l'IMC moyen (*créer un résultat sommaire*) des patientes ayant entre 25 et 30 ans (*sélectionner des observations*), et ce en comparant les patientes qui ont reçu le traitement à celles qui ne l'ont pas reçu (*créer des groupes*).

À vous de jouer! (2)

Action

Sélectionner des observations

Sélectionner des variables

Ordonner les lignes

Créer de nouvelles variables

Créer des résultats sommaires

Créer des groupes

Verbe

`filter(df, critère)`

`select(df, critère)`

`arrange(df, critère)`

`mutate(df, critère)`

`summarise(df, critère)`

`group_by(df, critère)`

RetouR

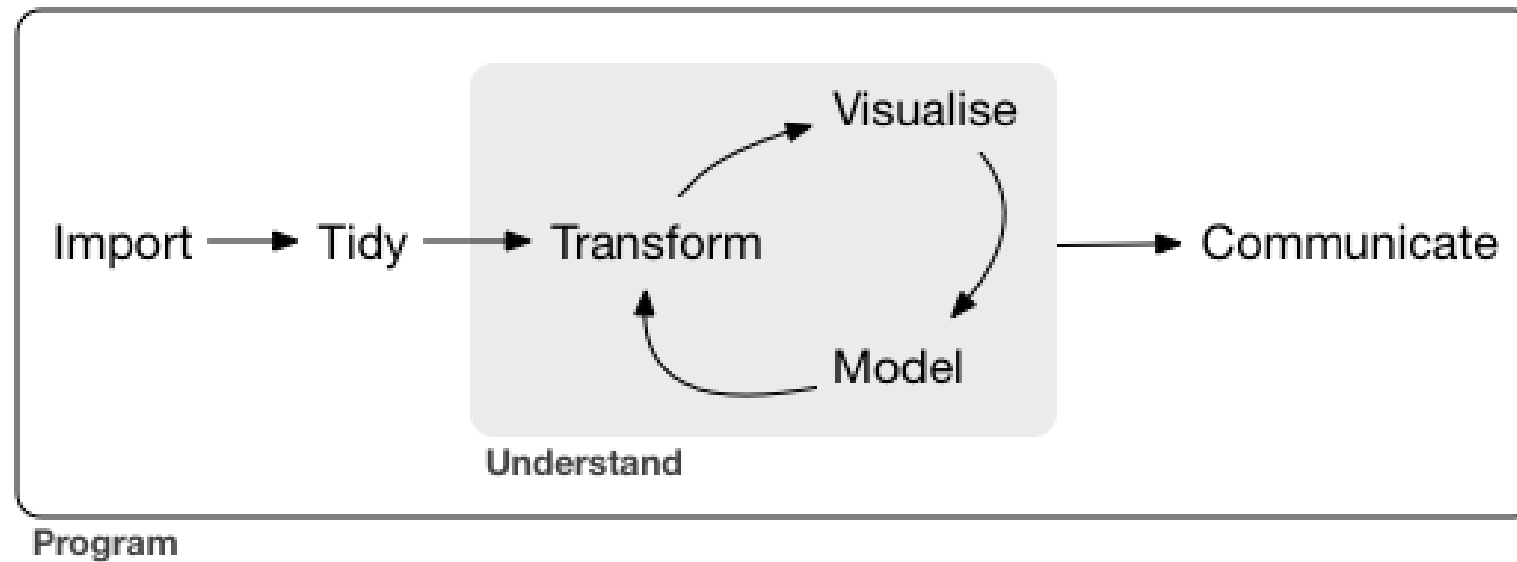
Création d'une variable catégorique:

```
df_glucose <- df_glucose %>%  
  mutate(AgeCat = case_when(Age < 25 ~ "<25",  
                             Age < 32 ~ "≥ 25, <32",  
                             Age ≥ 32 ~ "≥ 32"))
```

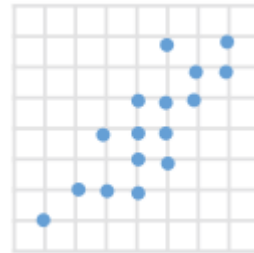
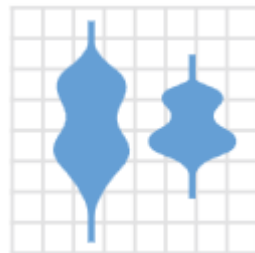
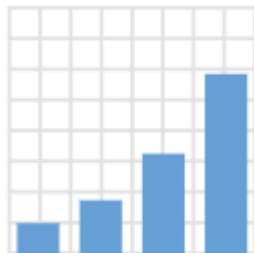
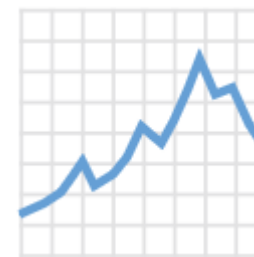
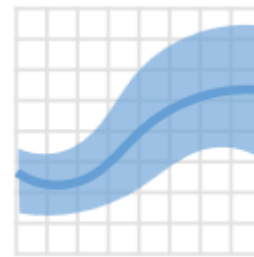
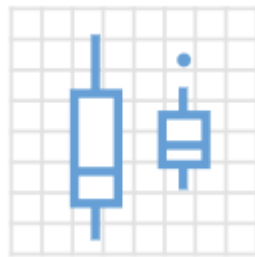
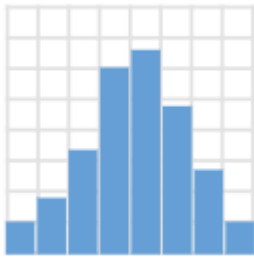
Calcul d'un compte:

```
df_glucose %>%  
  filter(Age > 30) %>%  
  count(Traitement)
```

Visualisation de données



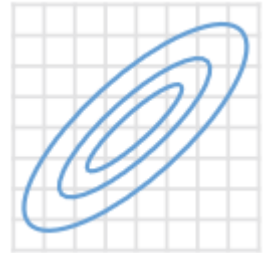
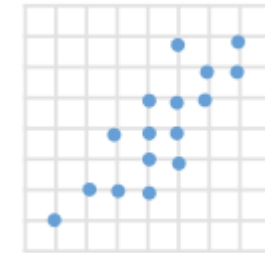
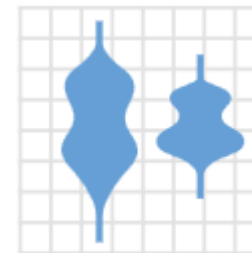
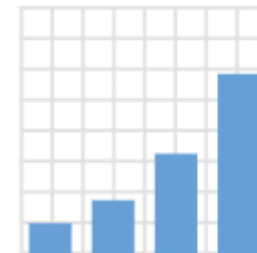
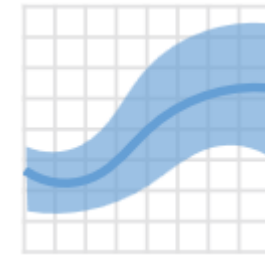
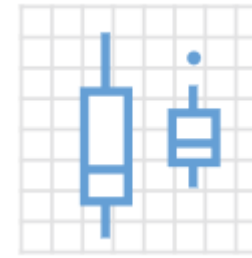
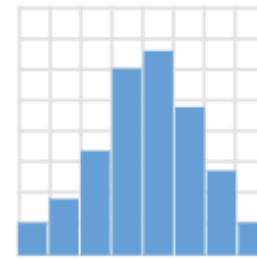
Visualisation de données



Visualisation de données

L'ensemble des graphiques que l'on crée partagent certains éléments essentiels:

- On a besoin d'un jeu de données;
- On doit choisir quelle(s) variable(s) on aimerait afficher;
- On doit choisir une méthode de représentation des données.



Visualisation de données

Éléments essentiels

Choix du jeu de données

Choix de la ou des variable(s)

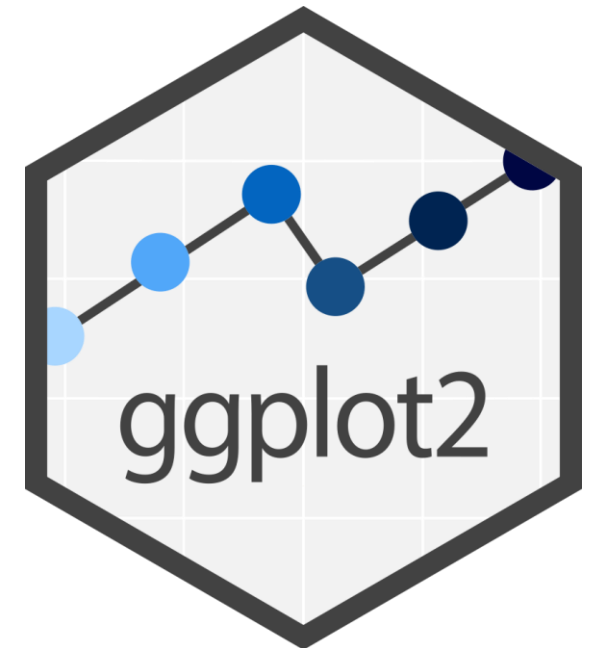
Choix de représentation

Fonction

`ggplot(df)`

`aes(x, y)`

`geom_ ... ()`



Visualisation de données

Éléments essentiels

Choix du jeu de données

Choix de la ou des variable(s)

Choix de représentation

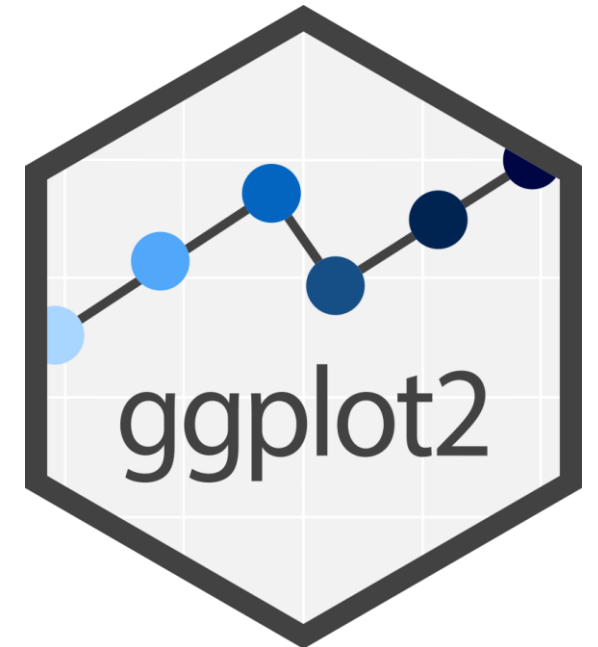
Fonction

`ggplot(df)`

`aes(x, y)`

`geom_ ... ()`

Le principe général de la *grammaire des graphiques*, c'est d'**ajouter** les éléments essentiels et optionnels d'un graphique *une couche à la fois*. Ceci rend les graphiques manipulables et malléables.



Visualisation de données (Exemple)

Exemples:

- Construire un histogramme de l'âge des patientes.

Visualisation de données (Exemple)

Exemples:

- Construire un histogramme (*représentation*) de l'âge (*variable*) des patientes.

Visualisation de données (Exemple)

Exemples:

- Construire un histogramme de l'âge des patientes ayant reçu le traitement et un autre pour celles ne l'ayant pas reçu.
- Construire un seul histogramme qui combine l'âge des patientes ayant reçu le traitement et celles ne l'ayant pas reçu.

Visualisation de données (Exemple)

Exemples:

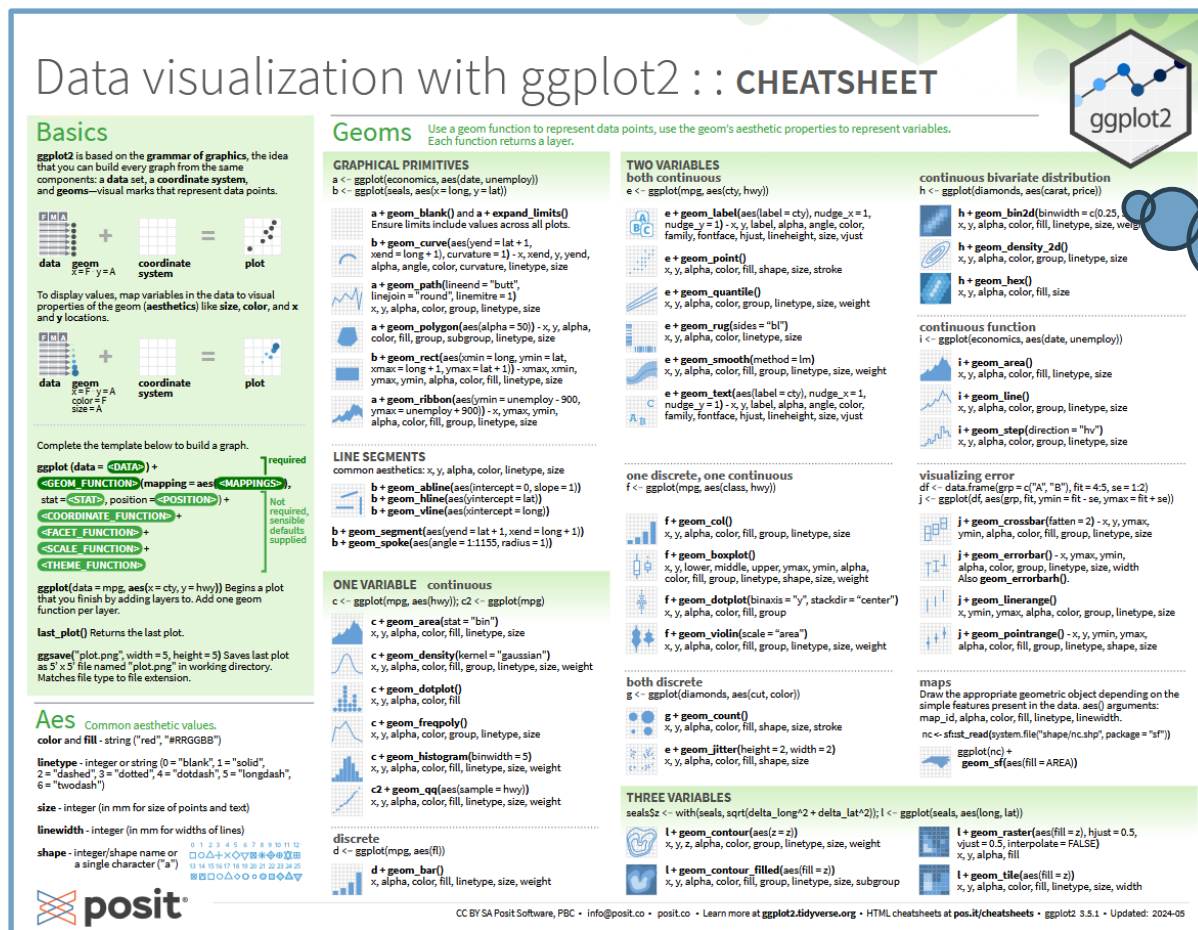
- Construire un histogramme (*représentation*) de l'âge (*variable*) des patientes ayant reçu le traitement et un autre pour celles ne l'ayant pas reçu (*choix de facette*).
- Construire un seul histogramme qui combine l'âge des patientes ayant reçu le traitement et celles ne l'ayant pas reçu.

Visualisation de données (Exemple)

Exemples:

- Construire un histogramme (*représentation*) de l'âge (*variable*) des patientes ayant reçu le traitement et un autre pour celles ne l'ayant pas reçu (*choix de facette*).
- Construire un seul histogramme (*représentation*) qui combine l'âge (*variable*) des patientes ayant reçu le traitement et celles ne l'ayant pas reçu (*choix d'étiquettes/d'esthétique*).

Visualisation de données... en trichant!



Les feuilles de triches sont disponibles dans le répertoire de l'atelier!

À vous de jouer! (3)

Éléments essentiels

Choix du jeu de données

Choix de la ou des variable(s)

Choix de représentation

Fonction

`ggplot(df)`

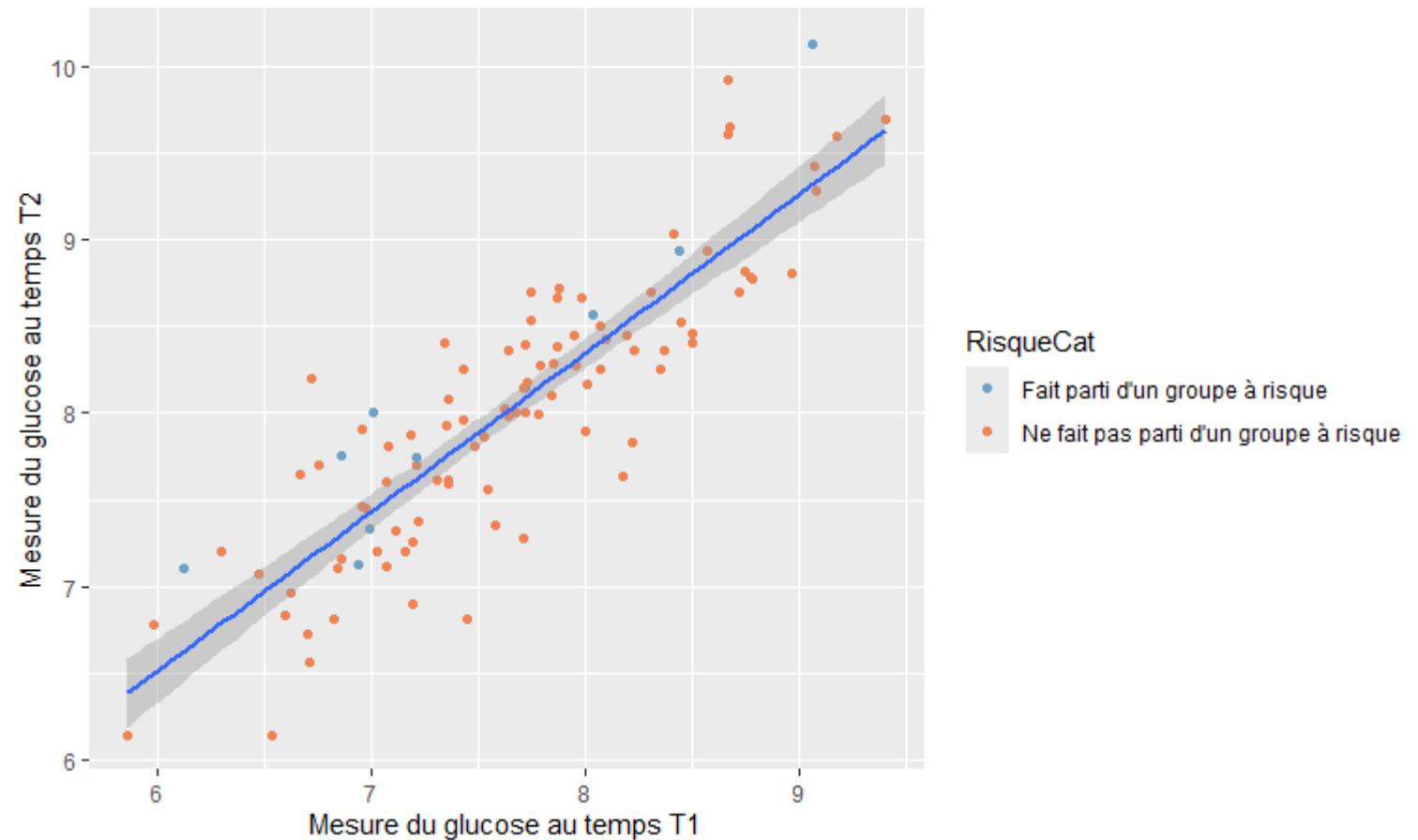
`aes(x, y)`

`geom_ ... ()`

Le principe général de la *grammaire des graphiques*, c'est d'ajouter les éléments essentiels et optionnels d'un graphique *une couche à la fois*. Ceci rend les graphiques manipulables et malléables.

RetouR

Étude de la relation entre les deux mesures du glucose en fonction de l'appartenance à un groupe à risque.





Exploration d'un jeu de données: *Données sur le diabète gestationnel*

Mise en situation

Le diabète gestationnel est associé à des conséquences négatives telles que l'obésité et l'intolérance au glucose chez l'enfant, ainsi qu'à des risques plus élevés de diabète de type 2 et de maladies cardiovasculaires pour la mère.

Dans la littérature, on remarque une association forte et continue entre les niveaux de glucose maternel du 3^e trimestre et l'augmentation du poids à la naissance ainsi que d'autres complications de la grossesse.

En cas de diabète gestationnel, le traitement usuel est de référer à l'équipe multidisciplinaire pour les grossesses à risque dans les hôpitaux.

Question de recherche

Une intervention en début de grossesse visant à améliorer la qualité de l'alimentation permet-elle d'améliorer le profil glycémique des personnes à risque de développer un diabète gestationnel?

Exploration du jeu de données

Avant de tenter de répondre directement à la question de recherche, une bonne pratique est d'explorer minimalement les données.

Dans ce cas-ci, on voudra:

- Jeter un coup d'œil aux statistiques descriptives du jeu;
- Comparer les groupes d'intérêt, s'il y a lieu.

À vous de jouer! (4)

On vous demande de comparer les statistiques descriptives du groupe « intervention » et du groupe « contrôle » pour les variables suivantes:

- Age
- Groupe_a_risque
- IMC
- Predia.

Pour une variable catégorique, on compare normalement uniquement les proportions. Pour une variable numérique, on compare normalement la moyenne, l'écart-type, la médiane (potentiellement avec les quartiles), le minimum et le maximum.

RetouR

Characteristic	Controle N = 51 ¹	Intervention N = 51 ¹	p-value ²
Age			<0.001
N Non-missing/No. obs. (% Non-missing%)	51.0/51.0 (100.0%)	51.0/51.0 (100.0%)	
Mean (SD)	27.2 (4.5)	30.6 (4.8)	
Median (Q1, Q3)	27.0 (24.0, 30.0)	30.0 (27.0, 35.0)	
Min, Max	17.0, 35.0	22.0, 39.0	
Groupe_a_risque	6 / 51 (12%)	4 / 51 (7.8%)	0.7
IMC			<0.001
N Non-missing/No. obs. (% Non-missing%)	51.0/51.0 (100.0%)	51.0/51.0 (100.0%)	
Mean (SD)	23.8 (2.0)	30.3 (2.0)	
Median (Q1, Q3)	23.8 (22.2, 25.1)	30.2 (29.0, 31.4)	
Min, Max	19.8, 29.7	26.1, 35.8	
PrediaCat			>0.9
Non	38 / 51 (75%)	39 / 51 (76%)	
Oui	13 / 51 (25%)	12 / 51 (24%)	

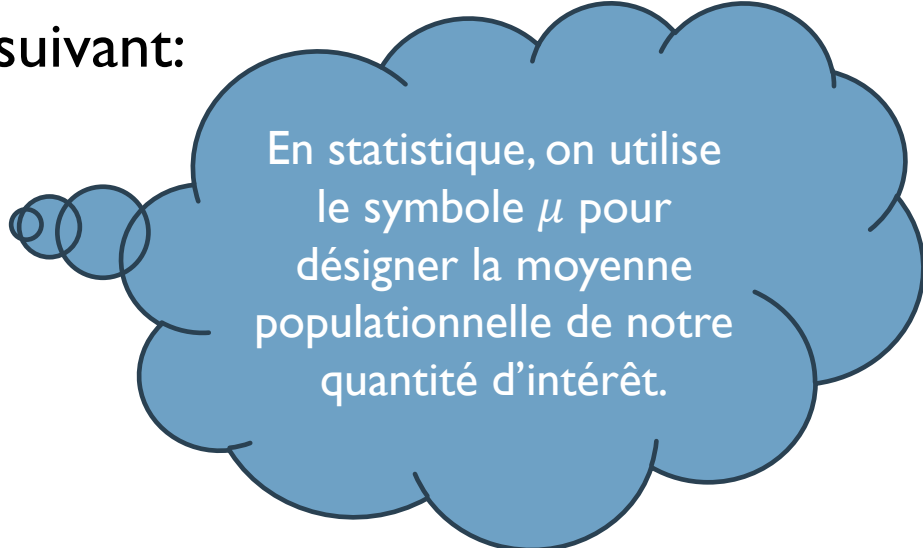
¹ n / N (%)

Notion de valeur- p

Étant donné un certain test d'hypothèses où l'on confronte l'hypothèse nulle H_0 avec l'hypothèse alternative H_1 , la valeur- p représente **la probabilité d'obtenir des résultats au moins aussi « extrêmes » que ceux que l'on a observés.**

Par exemple, on pourrait considérer le test d'hypothèses suivant:

$$\begin{cases} H_0: \mu_{\text{Contrôle}} = \mu_{\text{Intervention}} \\ H_1: \mu_{\text{Contrôle}} \neq \mu_{\text{Intervention}} \end{cases}$$



En statistique, on utilise le symbole μ pour désigner la moyenne populationnelle de notre quantité d'intérêt.

Notion de valeur- p (Exemple)

Characteristic	Controle N = 51 ¹	Intervention N = 51 ¹	p-value ²
Age			<0.001
N Non-missing/No. obs. (% Non-missing%)	51.0/51.0 (100.0%)	51.0/51.0 (100.0%)	
Mean (SD)	27.2 (4.5)	30.6 (4.8)	
Median (Q1, Q3)	27.0 (24.0, 30.0)	30.0 (27.0, 35.0)	
Min, Max	17.0, 35.0	22.0, 39.0	

Question de recherche, précisée!

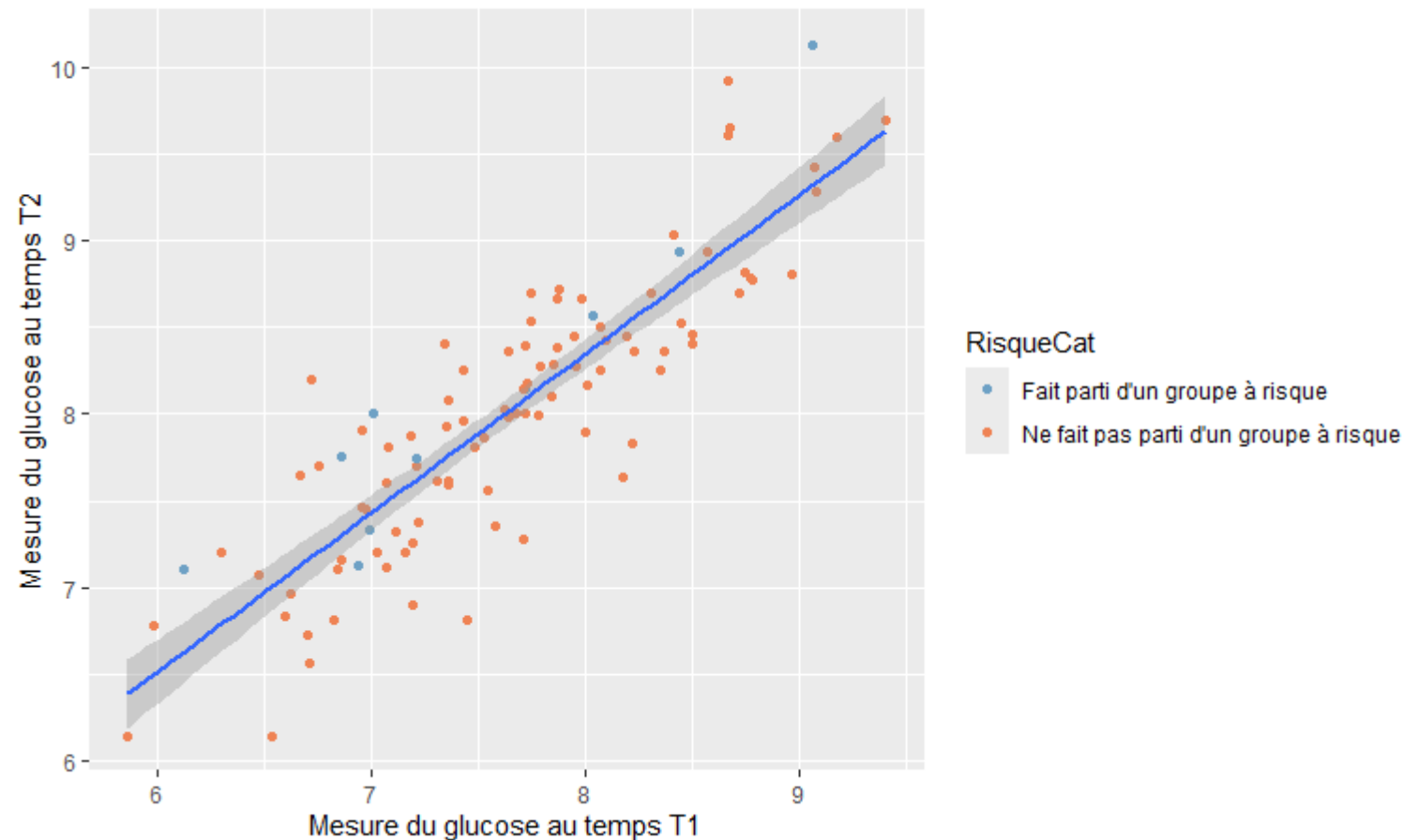
Une intervention en début de grossesse visant à améliorer la qualité de l'alimentation permet-elle d'améliorer le profil glycémique des personnes à risque de développer un diabète gestationnel?

Plus précisément, on dira qu'il y a eu une amélioration si l'augmentation dans la mesure du test oral pris au 3^e et au 1^{er} trimestre est inférieure à 0,3mmol/L.

Enfin, on aimerait savoir si la proportion de patientes ayant eu une amélioration est la même chez le groupe intervention que chez le groupe contrôle.

Survol du modèle de régression linéaire

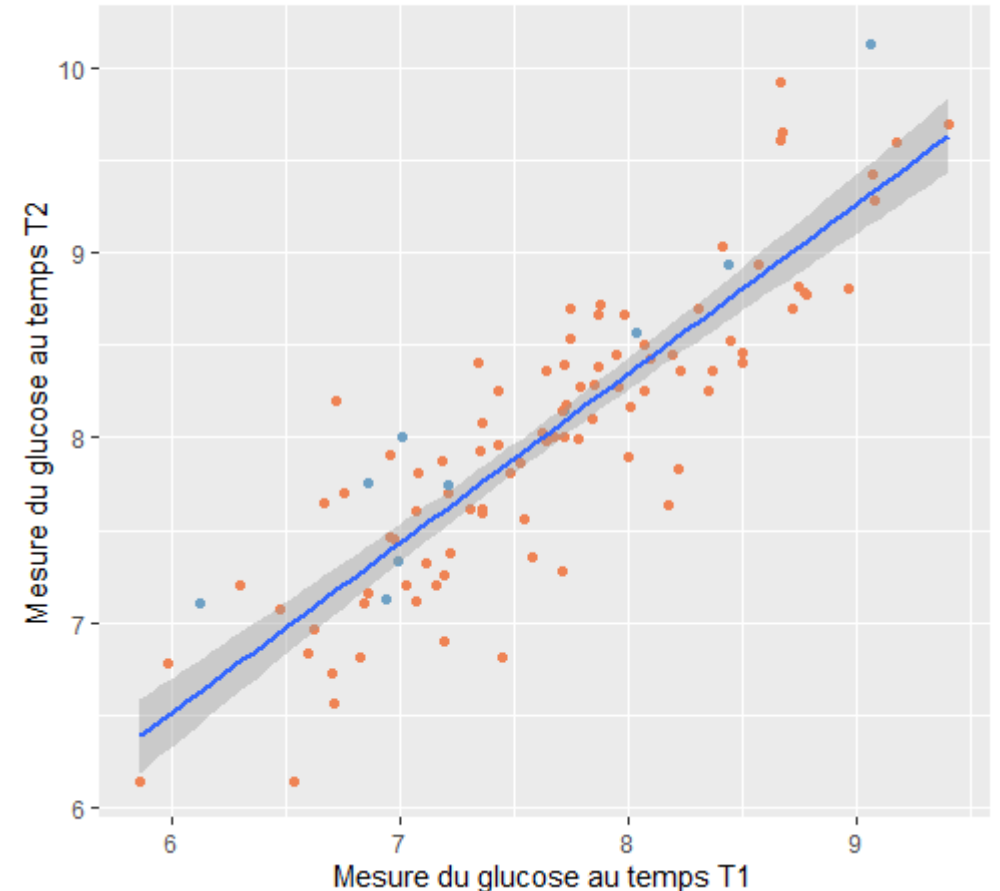
Étude de la relation entre les deux mesures du glucose en fonction de l'appartenance à un groupe à risque.



Survol du modèle de régression linéaire

- La variable d'intérêt, Y_i , est continue.
- Modélisation de la moyenne à l'aide d'une fonction linéaire: $\mathbb{E}[Y_i|x_i] = \beta_0 + \beta_1 x_i$.
- Interprétation des paramètres: ordonnée à l'origine (β_0) et pente (β_1).
- Hypothèses.
- Inférence.

Étude de la relation entre les deux mesures du glucose en fonction de l'appartenance à un groupe à risque.



Survol du modèle de régression linéaire (Prédicteurs cat.)

Lorsque l'on inclut des variables catégoriques (au format *factor*) dans un modèle linéaire à l'aide de R, la première catégorie devient ce qu'on appelle le groupe de référence. Les autres catégories seront quant à elles comparées au groupe de référence. On cherche alors à comprendre comment les autres catégories se distinguent, si c'est le cas, du groupe de référence.

Survol du modèle de régression linéaire (Prédicteurs cat.)

Lorsque l'on inclut des variables catégoriques (au format *factor*) dans un modèle linéaire à l'aide de R, la première catégorie devient ce qu'on appelle le groupe de référence. Les autres catégories seront quant à elles comparées au groupe de référence. On cherche alors à comprendre comment les autres catégories se distinguent, si c'est le cas, du groupe de référence.

La patiente fait partie de la catégorie d'âge...

- Moins de 25 ans
- Au moins 25 ans et moins de 32 ans
- Au moins 32 ans

Survol du modèle de régression linéaire (Prédicteurs cat.)

Lorsque l'on inclut des variables catégoriques (au format *factor*) dans un modèle linéaire à l'aide de R, la première catégorie devient ce qu'on appelle le groupe de référence. Les autres catégories seront quant à elles comparées au groupe de référence. On cherche alors à comprendre comment les autres catégories se distinguent, si c'est le cas, du groupe de référence.

La patiente fait partie de la catégorie d'âge...

- Moins de 25 ans → Groupe de référence
- Au moins 25 ans et moins de 32 ans → Groupe d'intérêt #1
- Au moins 32 ans → Groupe d'intérêt #2

Survol du modèle de régression linéaire (Inférence)

Pour les patientes de moins de 25 ans:

$$\hat{y}_{T_2} = 1 + 0,91x_{T_1}$$

Pour les patientes d'au moins 25 ans,
moins de 32 ans :

$$\hat{y}_{T_2} = 0,98 + 0,91x_{T_1}$$

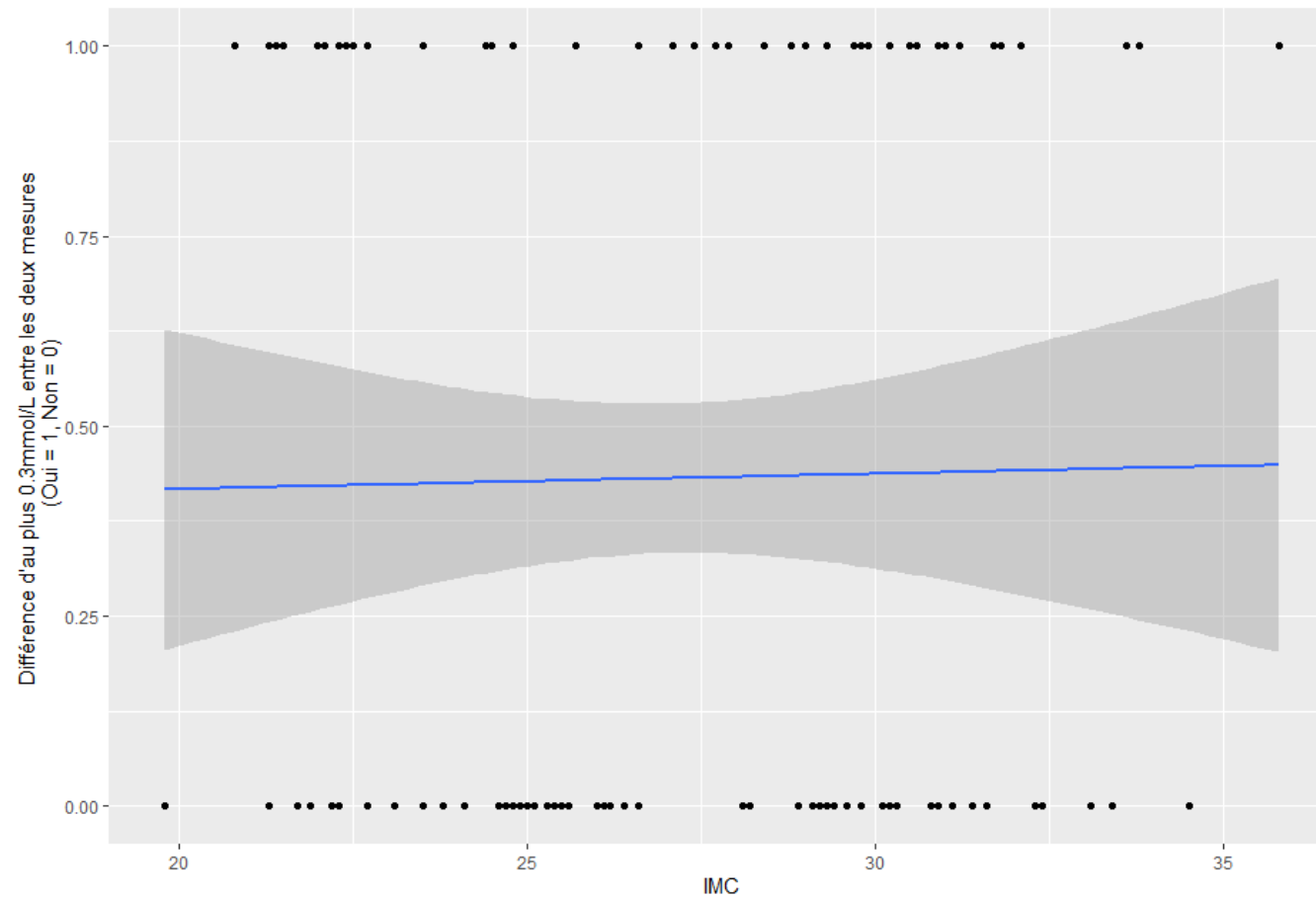
Pour les patientes d'au moins 32 ans:

$$\hat{y}_{T_2} = 1,11 + 0,91x_{T_1}$$

Characteristic	Beta	95% CI	p-value
(Intercept)	1.0	0.23, 1.9	0.013
Gluc_T1	0.91	0.81, 1.0	<0.001
AgeCat			
<25	—	—	
>=25, < 32	-0.02	-0.23, 0.19	0.9
>=32	0.11	-0.12, 0.35	0.3

Abbreviation: CI = Confidence Interval

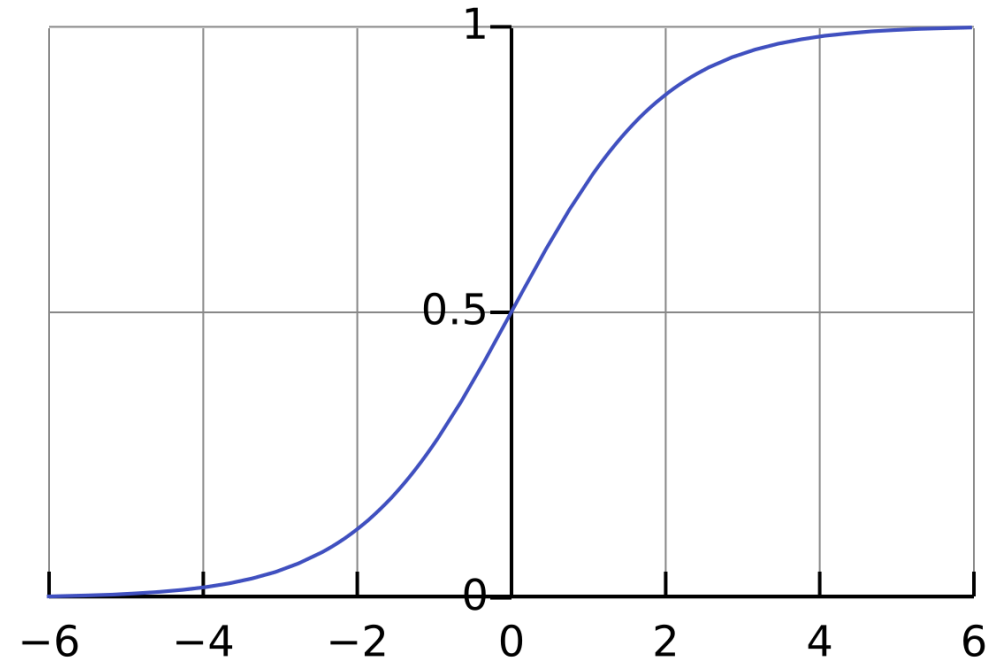
Introduction au modèle de régression logistique



Introduction au modèle de régression logistique

Particularités du modèle de régression logistique:

- La variable d'intérêt, Y_i , prend les valeurs 0 ou 1.
- Modélisation de la probabilité des deux réponses possibles:
 - $\mathbb{P}(Y_i = 1|X_i = x_i) = \frac{\exp(\beta_0 + \beta_1 x_i)}{1 + \exp(\beta_0 + \beta_1 x_i)}$
 - $\mathbb{P}(Y_i = 0|X_i = x_i) = \frac{1}{1 + \exp(\beta_0 + \beta_1 x_i)}$
- Interprétation des paramètres en termes de cotes (*odds*) ou de rapport de cotes (*odds ratio*).



Interprétation des paramètres (Exemple: prédicteur discret)

Modélisation

- Y : Avoir un cancer du poumon (oui = 1, non = 0)
- X : Fumer (oui = 1, non = 0)

Valeur de $\exp(\beta)$	Interprétation
$\exp(\beta) < 1$	Il est moins probable d'avoir le cancer du poumon si l'on fume que si l'on ne fume pas.
$\exp(\beta) = 1$	Il est autant probable d'avoir le cancer du poumon si l'on fume que si l'on ne fume pas.
$\exp(\beta) > 1$	Il est plus probable d'avoir le cancer du poumon si l'on fume que si l'on ne fume pas.

COTE

$$Cote(A) = \frac{\mathbb{P}(A)}{1 - \mathbb{P}(A)}$$

CAS OÙ $x_i = 0$

$$\frac{\mathbb{P}(Y_i = 1|X_i = 0)}{\mathbb{P}(Y_i = 0|X_i = 0)} = e^{\beta_0}$$

CAS OÙ $x_i = 1$

$$\frac{\mathbb{P}(Y_i = 1|X_i = 1)}{\mathbb{P}(Y_i = 0|X_i = 1)} = e^{\beta_0 + \beta_1}$$

RAPPORT DES COTES

$$e^{\beta_1} = \frac{e^{\beta_0 + \beta_1}}{e^{\beta_0}}$$

Interprétation des paramètres (prédicteur continu)

Dans le cas d'un prédicteur continu, le rapport des cotes du prédicteur X indique comment la cote va varier en fonction d'une augmentation d'une unité du prédicteur. Encore une fois, on suppose que les autres prédicteurs demeurent constants.

À noter qu'il s'agit d'un effet *multiplicateur* et non pas d'un effet additif.

Interprétation des paramètres (Exemple: prédicteur continu)

Modélisation

- Y : Avoir un cancer du poumon (oui = 1, non = 0)
- X : Âge de l'individu

Valeur de $\exp(\beta)$	Interprétation
$\exp(\beta) < 1$	Il est moins probable d'avoir le cancer du poumon lorsque l'âge augmente.
$\exp(\beta) = 1$	La probabilité d'avoir le cancer du poumon ne varie pas en fonction de l'âge.
$\exp(\beta) > 1$	Il est plus probable d'avoir le cancer du poumon lorsque l'âge augmente.

COTE

$$\text{Cote}(A) = \frac{\mathbb{P}(A)}{1 - \mathbb{P}(A)}$$

CAS OÙ $x_i = x$

$$\frac{\mathbb{P}(Y_i = 1 | X_i = x)}{\mathbb{P}(Y_i = 0 | X_i = x)} = e^{\beta_0 + \beta_1 x}$$

CAS OÙ $x_i = x + 1$

$$\frac{\mathbb{P}(Y_i = 1 | X_i = x + 1)}{\mathbb{P}(Y_i = 0 | X_i = x + 1)} = e^{\beta_0 + \beta_1 (x+1)}$$

RAPPORT DES COTES

$$e^{\beta_1} = \frac{e^{\beta_0 + \beta_1 (x+1)}}{e^{\beta_0 + \beta_1 x}}$$

Utilisation de R pour la régression logistique

La commande « `glm` » permet de construire un *generalized linear model* (qui correspond à une grande classe de modèle incluant la régression logistique). Il s'agit d'une commande disponible parmi les librairies de bases de R.

```
reg.logis ← glm( formula, family, data )
```

- **Formula:** Correspond au modèle qui nous intéresse.
On l'écrit: $y \sim x_1 + x_2 + \dots + x_p$
- **Family:** Permet de préciser quel GLM on souhaite utiliser.
On précise “binomial” pour la régression logistique.
- **Data:** Correspond au jeu de données que l'on souhaite utiliser.

Que peut-on dire par rapport à notre objectif?

Une intervention en début de grossesse visant à améliorer la qualité de l'alimentation permet-elle d'améliorer le profil glycémique des personnes à risque de développer un diabète gestationnel?

Plus précisément, on dira qu'il y a eu une amélioration si l'augmentation dans la mesure du test oral pris au 3^e et au 1^{er} trimestre est inférieure à 0,3mmol/L.

Enfin, on aimerait savoir si la proportion de patientes ayant eu une amélioration est la même chez le groupe intervention que chez le groupe contrôle.

Characteristic	OR	95% CI	p-value
TraitementCat			
Controle	—	—	
Intervention	6.68	1.27, 39.5	0.029
Age	0.92	0.84, 1.01	0.089
IMC	0.84	0.67, 1.03	0.11
Groupe_a_risque	0.12	0.01, 0.76	0.061
Predia	0.80	0.29, 2.17	0.7
Abbreviations: CI = Confidence Interval, OR = Odds Ratio			

Conclusion

Introduction à R avec un cas d'étude

```
Contexte(R) %>%  
  Particularités() %>%  
  RStudio() %>%  
  Tidyverse() %>%  
  ggplot2() %>%  
  CaseStudy() %>%  
  glm() %>%  
  Conclusion()
```

Références

- Ihaka, R., & Gentleman, R. (1996). R: a language for data analysis and graphics. *Journal of computational and graphical statistics*, 5(3), 299-314.
- Statistics Team. (s.d.). *Introduction to R for Health Data Science*. Repéré en ligne le 15 avril 2025, à partir de https://bookdown.org/m_p_sperrin/introduction_to_r/https://r4ds.had.co.nz/index.html
- Wickham, H., Çetinkaya-Rundel, M., & Grolemund, G. (2023). *R for Data science: Import, Tidy, Transform, Visualize, and Model Data*. O'Reilly Media
- Wickham, H. (2016). ggplot2. In Use R! <https://doi.org/10.1007/978-3-319-24277-4>
- Boehmke, B. C., PhD. (2016). *Data Wrangling with R*. Springer.
- Peng, R. D. (2012). *R Programming for data science*.