

Cette présentation a été effectuée le 29 novembre 2023, au cours de la journée Le logiciel R : un outil pour optimiser la vigie-surveillance dans le cadre des 26es Journées annuelles de santé publique.



Le logiciel R : un outil pour optimiser la vigie-surveillance

Personnes formatrices et
membres du comité scientifique

29 novembre 2023

Le temps des
concordances

Plan de l'atelier

1. Mot d'introduction, objectifs et déroulement
2. Mise en contexte et discussion 1
3. Logiciel R, avantages et limites, quelques notions
4. Présentations et exercices 1-4
5. Discussion 2
6. Conclusion

Mot d'introduction, objectifs et déroulement



Mot d'introduction



- ▶ Pourquoi un atelier sur l'utilisation du logiciel R en vigie-surveillance?
 - Démocratisation et collaboration
- ▶ Personnes formatrices : Angélique, Éric, Flavie (responsable), Kevin, Michèle-Victoria et Romain (coresponsable)
 - Expérience avec R : variable

Mot d'introduction

- ▶ Retour sur le sondage soumis aux personnes participantes
 - 28 réponses
 - 82 % n'ont jamais utilisé R ou débutent
 - 68 % ont R sur l'ordinateur de leur organisation
 - 25 % ont utilisé R en milieu professionnel
 - 68 % ont le titre d'emploi APPR ou conseiller scientifique

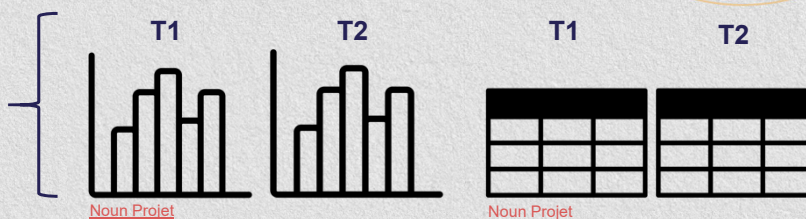
Objectifs

- ▶ Utiliser le logiciel R pour effectuer certaines tâches essentielles (ex. importer des données, télécharger les packages, exporter des données et des graphiques/tableaux);
- ▶ Manipuler des sections de codes R et des packages utiles au traitement de données ainsi qu'à la création et à la mise à jour automatisée de tableaux et de graphiques;
- ▶ Identifier les avantages et les limites du logiciel R.

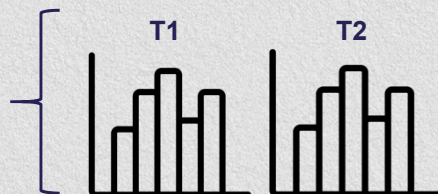
Objectifs



Liste de cas fictive, Montérégie



Indicateur agrégé, ensemble du Québec



Temps 1 (T1) :
2022-09-25 au 2023-09-16

Temps (T2) :
2022-09-25 au 2023-09-23

Objectifs

- ▶ Créer un graphique d'une **courbe épidémique** du nombre de cas de streptocoque du groupe A (SGA) par semaine CDC en **Montérégie** et l'actualiser*
- ▶ Créer un **tableau** des cas de SGA en fonction du **statut d'hospitalisation**** en Montérégie et l'actualiser
- ▶ Créer un graphique d'une **courbe épidémique** du nombre de cas de SGA pour l'**ensemble du Québec** et l'actualiser

*Ajout d'une semaine de données (correspond au "T2")

**Données fictives

Déroulement

- ▶ Alternance : présentations, exercices pratiques, discussions
- ▶ Durant les exercices
 - animateurs et participants exécutent le code en simultané
 - Moments planifiés pour les questions
 - Code de l'atelier est parsemé d'exercices
 - animateurs répondent aux questions et fournissent les solutions aux exercices dans la boîte de clavardage
 - Retour sur l'exercice

```
# Exercice pratique 2.2
#
# Créer une variable "gr_age_v2" dont l
## à partir de la variable age

sga_2_2 <- sga %>%

sga <- sga_2_2
#
```

Déroulement

- ▶ Convivialité, entraide et respect de l'horaire établi
 - Code détaillé avec commentaires, utile pour étudier le code après l'atelier
 - Périodes "tampons" : retour sur l'exercice X, *Pour aller plus loin*
- ▶ Options pour poser des questions
 - Adresser les questions dans la boîte de clavardage
 - Allumer sa caméra

Déroulement

► Gestion des pépins - quelques exemples

Pépin	Solution
Difficulté à importer les données ou un package	Se joindre à la table "problèmes techniques logiciel R"
Message d'erreur lors de l'importation de données ... <i>does not exist in current working directory ('C:/Users/MF/...)</i>	Fermer RStudio et ouvrir le projet "Atelier_JASP" à partir du dossier. Ensuite, accéder aux scripts
Les caractères spéciaux (ex. "é") ne s'affichent pas bien	<i>RStudio > File > Reopen with Encoding... > ISO-88591-1 OU UTF-8 > Ok</i>
Message d'erreur dans la console	Exécuter le code depuis le début
Manque de temps pour compléter un exercice	Copier le code soumis dans la boîte de clavardage dans son code
Difficulté à rattraper un retard	Utiliser le code complet et l'exécuter

Discussion 1

Pourquoi avoir choisi R?

► Contexte

- Production fréquente de portraits épidémiologiques COVID-19
- Plusieurs figures à actualiser dans un fichier Excel volumineux

► Avantages R

- Logiciel gratuit, communauté de pratique
- Réduction du nombre de manipulations et du temps de production
- Innovation (graphique dynamique, rapport avec R Markdown)

► Défis R

- Temps associé à une 1ère production (programmation)

Votre expérience

► Quels sont les outils utilisés présentement pour produire des données épidémiologiques de façon fréquente (ex. COVID-19)?

- Quels sont les enjeux rencontrés?
- Envisagez-vous de changer ou d'optimiser vos outils de vigie-surveillance (hebdomadaire/mensuel/annuel)?



Logiciel R, avantages et limites, quelques notions



Logiciel R, avantages et limites

► R

- Langage de programmation et logiciel libre d'accès
- Permet traitement, analyse et visualisation de données

► RStudio

- Environnement de développement
- Permet l'utilisation du langage R et d'autres outils

Avantages	<ul style="list-style-type: none">• Gratuit et libre d'accès• Outils de traitement de données• Diversité et personnalisation de graphiques/tableaux• Transparence et reproductibilité• Analyses statistiques complexes• Rapport avec R Markdown
Limites	<ul style="list-style-type: none">• Courbe d'apprentissage abrupte• Discipline pour structurer le code
Les deux	<ul style="list-style-type: none">• Utilisation de packages• Mises à jour fréquentes• Diversité de "chemins"

Quelques comparaisons

Table 1. Comparison of statistical softwares [6]

Software	Advantages	Disadvantages
Excel	Free Simple to use	Limited number of tests available
SPSS	Graphical interface Easy to learn, intuitive Large number of available statistical tests	Expensive Limited options for transformation of data
SAS	Can process extremely large datasets Can perform advanced analyses	Script based, so it demands considerable learning Expensive
R	Free Can perform advanced analyses	Script based, so it demands considerable learning
STATA	Has both graphical and script interface Can perform advanced analyses	Expensive More difficult to learn than SPSS

[Lien vers l'article](#)

Utilisation complémentaire

Power BI (Outil de visualisation)	Simple d'utilisation Permet de personnaliser la visualisation	Coût/Accès (en 2020-2021) Options limitées pour la transformation de données
-----------------------------------	--	---

Interface

The screenshot shows the RStudio interface with the following components highlighted:

- Source/Éditeur (Fichiers de travail (codes, bd))**: The main editor window showing R code for package installation and library loading.
- Environnement (Objet créé)**: The Environment pane on the right showing the objects created in the current session.
- Console ("Résultat" du code)**: The Console pane at the bottom showing the output of the R code, including the R license text.

Termes importants

Nom	Fonction	Exemples
Objet	Stocke de l'information et est utilisé dans les analyses	Résultat de calcul, banque de données, liste de noms, liste de nombres, fonction
Type d'objet <i>Aussi appelé "classe"</i>	Indique les valeurs qu'un objet peut prendre. Ex.: numérique (<i>double</i>), entier/ <i>integer</i> , caractère, logique, facteur (<i>integer</i>), date (<i>double</i>)	134.55, 354, "atteinte neurologique", FALSE, "Oui", "2023-03-03"
Opérateur	Permet d'effectuer des opérations sur des objets ou valeurs	Assigner : <-, additionner : +, "ou" :
Fonction	Contient des extraits de code prêts à l'emploi	mean(nb_cas)
Package	Contient un ensemble de fonctions pour accomplir certaines tâches et de la documentation	base, ggplot2, gt
Script R/code	Fichier contenant le langage R et commentaires	01_nettoyer.R
Script R Markdown	Fichier contenant le langage R, commentaires et extraits. Permet de créer un rapport	Exemples de rapport R Markdown

Objet

- ▶ Élément qui stocke de l'information et qui est utilisé dans les analyses
- ▶ On s'y réfère par son nom (chaîne de caractères), **sans guillemets**
- ▶ Nom d'un objet
 - Ex :

```
> nom_resp_atelier <- "Flavie"
> nom_resp_atelier
[1] "Flavie"
```

 - Court et informatif
 - Commence par une lettre et peut contenir lettres, chiffres, ".", "_"
 - Astuces
 - Être consistant
 - Porter attention aux minuscules/majuscules (test ≠ Test)
 - Éviter d'utiliser le même nom plus d'une fois

Opérateur

► Opérateurs de base :

Arithmétique		Opérateurs Comparaison		Logique	
+	addition	<	inférieur à	! x	NON logique
-	soustraction	>	supérieur à	x & y	ET logique
*	multiplication	<=	inférieur ou égal à	x && y	idem
/	division	>=	supérieur ou égal à	x y	OU logique
^	puissance	=	égal	x y	idem
%%	modulo	!=	différent	xor(x, y)	OU exclusif
%%/	division entière				

[Lien vers le livre](#)

► *Pipe operator* de **tidyverse** : %>%

- Permet d'appliquer plusieurs fonctions (manipulations) subséquentes sur une banque de données
- Le résultat d'une fonction est utilisé comme point de départ pour la suivante

Tidyverse comprend un ensemble de packages et de fonctions servant à l'analyse de données

Fonction

► Objet contenant une série de commandes prêtes à l'emploi. Améliore la lisibilité et évite les "copier-coller" d'une même section de code

► Utilisation

- Définir les valeurs (appelées arguments) que prennent les **paramètres** de la fonction
- Si les valeurs par défaut des paramètres conviennent, on peut les omettre

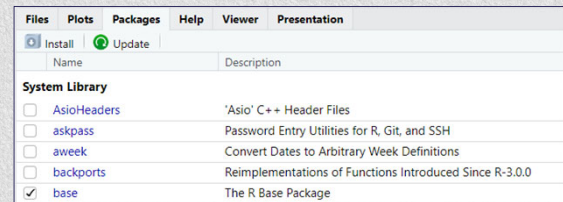
► Ex. mean() pour le calcul d'une moyenne

- mean(x, trim = 0, na.rm = FALSE, ...)

```
> serie <- c(2,3,4,5,6,6,7,3,3)
> mean(serie)
[1] 4.333333
```

Package

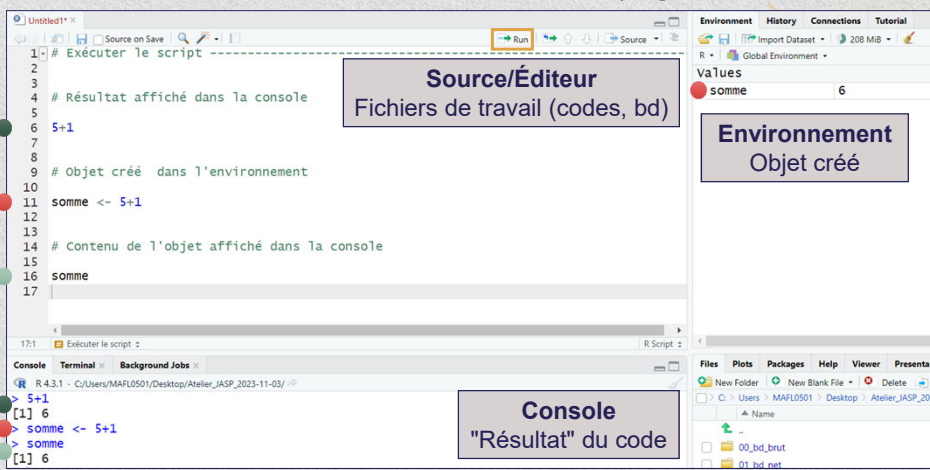
- ▶ Dossier incluant un ensemble de fonctions pour accomplir certaines tâches et de la documentation
- ▶ Comprend codes, documentation du package/fonctions, historique des versions/changements, banques de données, etc.
- ▶ Certains packages sont intégrés d'emblée dans R/RStudio (ex. **base**, **utils**, **graphics**); les autres doivent être installés (une fois) et appelés (chaque script)
- ▶ Utilisation
 - Installer : `install.packages("gt")`
 - Télécharger/appeler : `library(gt)`
 - Obtenir de l'aide : `?gt`
 - Cibler une fonction d'un package précis : `gt::tab_caption()`



Name	Description
System Library	
<input type="checkbox"/> AsioHeaders	'Asio' C++ Header Files
<input type="checkbox"/> askpass	Password Entry Utilities for R, Git, and SSH
<input type="checkbox"/> aweek	Convert Dates to Arbitrary Week Definitions
<input type="checkbox"/> backports	Reimplementations of Functions Introduced Since R-3.0.0
<input checked="" type="checkbox"/> base	The R Base Package

Script R – exécuter le code

- ▶ Se positionner sur la ligne de code > Run / "Ctrl+Enter"
- ▶ Sélectionner une section de code > Run / "Ctrl+Enter"



Source/Éditeur
Fichiers de travail (codes, bd)

Environnement
Objet créé

Console
"Résultat" du code

```
1 # Exécuter le script
2
3
4 # Résultat affiché dans la console
5
6 5+1
7
8
9 # Objet créé dans l'environnement
10
11 somme <- 5+1
12
13
14 # Contenu de l'objet affiché dans la console
15
16 somme
17
```

```
R 4.3.1 - C:/Users/MAFL0501/Desktop/Atelier_JASP_2023-11-03/
> 5+1
[1] 6
> somme <- 5+1
> somme
[1] 6
```

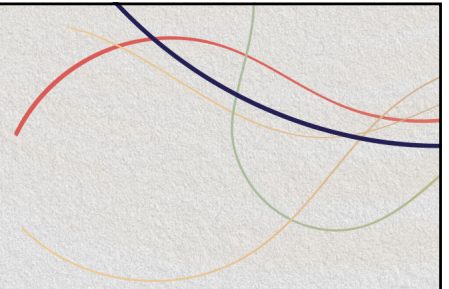
Script R – les bonnes pratiques

- ▶ Créer les scripts R à l'intérieur d'un **projet R**
 - Dossier incluant tous les fichiers reliés à un "projet" (scripts, données, figures)
 - Ce dossier peut être partagé et son contenu utilisé sans avoir à changer les répertoires de travail; ils sont relatifs au dossier du projet
- ▶ Débuter un script par une description et télécharger les packages nécessaires
- ▶ Fragmenter le script en sections et le commenter
- ▶ Privilégier le dialecte "tidy" plutôt que "base R" pour débiter (un débat)

Script R – accéder à une ligne/colonne

Action	R de base	Tidy
Accéder à une colonne par son nom	<pre>banque_donnee\$nom_col # Ou banque_donnee[, "nom_col"]</pre>	<pre>banque_donnee %>% select(nom_col)</pre>
Accéder à une ligne en fonction d'un critère	<pre>banque_donnee [banque_donnee\$nom_col == 10,]</pre>	<pre>banque_donnee %>% filter(nom_col == 10)</pre>

Questions



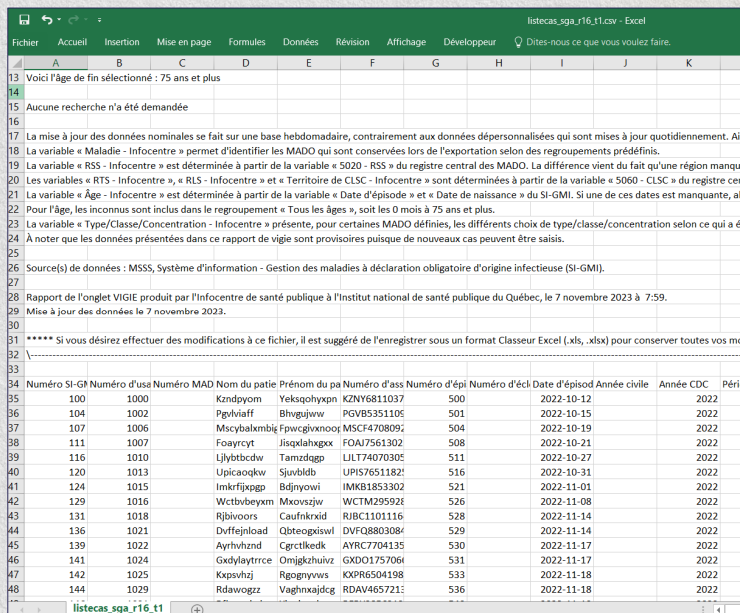
Présentation 1 : Importation et visualisation de données



Exercice 1 – un aperçu

- ▶ Se familiariser avec les banques de données (bd)
 - Liste de cas de SGA fictive, Montérégie
 - Respect confidentialité : variables dénominalisées, retirées, modifiées, mélangées
 - .CSV
 - Indicateur agrégé du nombre de cas de SGA, ensemble du Québec
 - Retrait des données non utilisées
 - .XLSX

Liste de cas de SGA fictive, Montérégie



Número SI-GM	Número d'usa	Número MAD	Nom du patie	Prénom du pa	Número d'ass	Número d'épi	Número d'écl	Date d'épisod	Année civile	Année CDC	Périod
100	1000	Kzndpyom	Yeksqohyxp	KZNY6811037	500			2022-10-12	2022		
104	1002	Pgvviiaff	Bhvgujwv	PGVB5351105	501			2022-10-15	2022		
107	1006	Mscybalkmbig	Fpwcgixvnoo	MSCF4708091	504			2022-10-19	2022		
111	1007	Foayrcyt	Jisqxlaxgxc	FOAI7561302	508			2022-10-21	2022		
116	1010	Ljybtbcdw	Tamzaggp	LJLT74070305	511			2022-10-27	2022		
120	1013	Upicaogkw	Sjvvlbdb	UPIS76511821	516			2022-10-31	2022		
124	1015	Imkrripqggp	Bdphyyowil	IMKB1853302	521			2022-11-01	2022		
129	1016	Wctvbyeyym	Mxvovsjw	WCTM29592E	526			2022-11-08	2022		
131	1018	Rjbtvoors	Caufnkxoid	RIBC11011116	528			2022-11-14	2022		
136	1021	Dvffejnload	Qlbtceogiswl	DVFO8803084	529			2022-11-14	2022		
139	1022	Ayrhvhznd	Cgrctlkedk	AYRC7704135	530			2022-11-17	2022		
141	1024	Gxdylaytrce	Omijgkhuivz	GXD01757061	531			2022-11-17	2022		
142	1025	Kxpsvhzj	Rgagnyvvvs	KXPR6504198	533			2022-11-18	2022		
144	1029	Rdawogzz	Vaghnxajdcg	RDAV4657211	536			2022-11-18	2022		

Le contenu des lignes diffère complètement de celui de la banque de données originale

Indicateur agrégé du nombre de cas de SGA, ensemble du Québec



Nombre de cas déclarés, du 25 septembre 2022 au 16 septembre 2023, infection à streptocoques groupe A, ensemble du Québec

Semaine CDC	Période	Nombre de cas
5	2022-2023	5
6	2022-2023	4
7	2022-2023	12
8	2022-2023	8
9	2022-2023	6
10	2022-2023	14
11	2022-2023	8
12	2022-2023	19
13	2022-2023	11
14	2022-2023	18
15	2022-2023	17
16	2022-2023	20
17	2022-2023	18
18	2022-2023	32
19	2022-2023	15
20	2022-2023	25
21	2022-2023	20
22	2022-2023	25
23	2022-2023	20
24	2022-2023	20
25	2022-2023	26
26	2022-2023	22
27	2022-2023	17
28	2022-2023	28
29	2022-2023	19
30	2022-2023	16
31	2022-2023	21
32	2022-2023	33
33	2022-2023	37
34	2022-2023	31
35	2022-2023	33
36	2022-2023	20
37	2022-2023	20
38	2022-2023	25

Exercice 1 – un aperçu

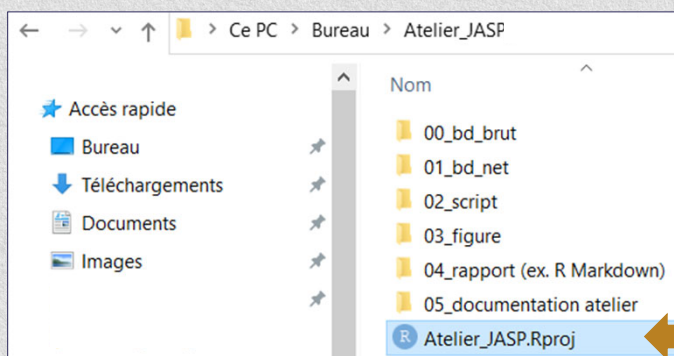
- ▶ Ouvrir le projet dans R
- ▶ Afficher le répertoire de travail
- ▶ Appeler les packages (installés avant l'atelier)
- ▶ Utiliser des fonctions servant à la visualisation
- ▶ Importer et visualiser les données de l'atelier
- ▶ **Exercice 1.1** : Inspecter les données téléchargées à l'aide des fonctions de visualisation

Exercice 1 – quelques fonctions

Package	Fonction	Utilité
base	getwd()	Afficher le répertoire de travail/ <i>working directory</i>
utils	View()	Afficher les données en "feuille de calcul" dans un onglet de l'éditeur
utils	head()	Afficher les premières lignes et colonnes d'une banque de données (bd) dans la console
read_r	read_delim()	Lire un fichier dont le contenu est séparé par des virgules/points-virgules (.csv)
readxl	read_excel()	Lire un fichier dont le format est .xls ou .xlsx
dplyr	glimpse()	Afficher une banque de données transposée de façon à voir toutes les colonnes et une partie du contenu des lignes dans la console
dplyr	tibble()	Créer une banque de données

Exercice 1 – documentation

► Structure du projet R "Atelier_JASP"



Cliquer ici pour ouvrir le projet

► Certains dossiers sont vides


Exercice 1 – clé pour lire les à, é, ï...

- ▶ Pour lire un fichier CSV, c.-à-d. fichier texte dont les valeurs sont séparées par une virgule/point-virgule, il faut spécifier l'encodage Unicode*
- ▶ Unicode : norme spécifiant la façon de coder les caractères spéciaux (ex. accents français)
- ▶ Un fichier texte peut être encodé de différentes façons (UTF-8, Latin-1, etc.)
- ▶ Les fichiers CSV de l'Infocentre utilisent l'encodage Latin-1, aussi appelé ISO-8859-1
- ▶ **L'Unicode ISO-8859-1 doit être spécifié pour lire la *Liste de cas de SGA fictive, Montérégie* dans R**

**Ceci ne s'applique pas aux fichiers Excel; ce ne sont pas des fichiers textes*

Questions

Exercice 1 : Importation et visualisation de données



Exercice 1



- ▶ Travail avec les animateurs
- ▶ Retour sur l'exercice (5 minutes)
 - Enjeux/bons coups au niveau du fonctionnement
 - Concepts à clarifier
 - Éléments intéressants (ex. échanges, solutions alternatives) à partager



Présentation 2 : Traitement et exportation de données



Exercice 2 – un aperçu

- ▶ Liste de cas fictive - Montérégie
 - Changer le nom des variables
 - Sélectionner et modifier les variables d'intérêt
 - Compter et illustrer le nombre de cas par jour et par semaine CDC
 - **Exercice 2.1** : Compter le nombre de cas par année CDC et semaine CDC
 - Déterminer la date de début de semaine CDC correspondant à la date d'épisode
 - Manipuler la variable "age" utile pour le tableau des caractéristiques des cas
 - **Exercice 2.2** : Créer une variable "gr_age_v2" dont les catégories sont 0-18, 19-39, 40-64, ≥65 ans
 - *Pour aller plus loin* : créer et modifier les niveaux d'une variable de type facteur

Exercice 2 – un aperçu

- ▶ Indicateur agrégé - Ensemble du Québec
 - Traiter et illustrer les données
- ▶ Exportation des données nettoyées
 - Exporter la liste de cas
 - **Exercice 2.3** : Exporter les données agrégées

Exercice 2 – Concept de semaine CDC

- ▶ Basée sur la logique de la numérotation ISO des semaines
 - Semaines numérotées de 1 à 52 ou 1 à 53 dont le début est le **dimanche** (au lieu du lundi)
- ▶ Permet de compter de façon standardisée les cas d'une année à l'autre
- ▶ Durant l'atelier, il faudra
 - À partir d'une date (ex. 2023-06-01), déterminer la semaine CDC (ex. 2023-22)
 - À partir d'une semaine CDC (ex. 2023-22), déterminer la date correspondant au début d'une semaine CDC (ex. 2023-05-28)

Exercice 2 – quelques fonctions

Package	Fonction	Utilité
base	class()	Afficher la classe/type d'un objet (ex. une variable dans une bd)
base	as.Date()	Convertir la classe/type d'un objet en "date"
base	table()	Créer un tableau de contingence
graphics	plot()	Créer un graphique de base du type nuage de points
janitor	clean_names()	Nettoyer les noms des colonnes/variables pour qu'elles contiennent uniquement des lettres, chiffres et "_"
dplyr	mutate()	Créer, modifier ou supprimer une colonne(s)
dplyr	group_by()	Cibler une colonne(s) sur laquelle des opérations par groupe d'observation/ligne auront lieu
dplyr	summarise()	Créer une bd qui contient chaque groupe d'observation/ligne d'une colonne(s) et un sommaire statistique
lubridate	floor_date()	Arrondir un objet "date" à la borne inférieure de l'unité spécifié
readr	write_csv()	Créer un fichier .csv à partir d'une banque de données
writexl	write_xlsx()	Créer un fichier .xlsx à partir d'une banque de données

Exercice 2 – les fonctions dplyr

- ▶ Utiles pour le traitement de données
- ▶ Nom de chaque fonction correspond à son rôle
 - Ligne : filter(), arrange(), slice()
 - Colonne : rename(), relocate(), select(), mutate()
- ▶ [Cheatsheet](#)
- ▶ Rappel : l'opérateur %>% permet d'effectuer des manipulations subséquentes sur une banque de données

Data transformation with dplyr : : CHEATSHEET

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

Each variable is in its own column & Each observation, or case, is in its own row

x |> f(y) becomes f(x, y)

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).

summary function

Compute table of summaries.

```
mtcars |> summarize(avg = mean(mpg))
```

Count number of rows in each group defined by the variables in ... Also **tally()**, **add_count()**, **add_tally()**.

```
mtcars |> count(city)
```

Group Cases

Use **group_by()** of a 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.

```
mtcars |> group_by(cyl) |> summarize(avg = mean(mpg))
```

Manipulate Cases

EXTRACT CASES

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

filter() (data, ..., preserve = FALSE) Extract rows that meet logical criteria.

```
mtcars |> filter(mpg > 20)
```

distinct() (data, ..., keep_all = FALSE) Remove rows with duplicate values.

```
mtcars |> distinct(gear)
```

slice() (data, ..., preserve = FALSE) Select rows by position.

```
mtcars |> slice(10:15)
```

slice_sample() (data, ..., n, prop, weight_by = NULL, replace = FALSE) Randomly select rows. Use n to select a number of rows and prop to select a fraction of rows.

```
mtcars |> slice_sample(n = 5, replace = TRUE)
```

slice_min() (data, order_by, ..., n, prop, with_ties = TRUE) and **slice_max()** Select rows with the lowest and highest values.

```
mtcars |> slice_min(mpg, prop = 0.25)
```

slice_head() (data, ..., n, prop) and **slice_tail()** Select the first or last rows.

```
mtcars |> slice_head(n = 5)
```

Logical and boolean operators to use with filter()

```
== < <= is.na() %in% | xor()
!= > >= is.na() ! &
```

See ?base:Logic and ?Comparison for help.

ARRANGE CASES

Manipulate Variables

EXTRACT VARIABLES

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

pull() (data, var = 1, name = NULL, ...) Extract column values as a vector, by name or index.

```
mtcars |> pull(wt)
```

select() (data, ...) Extract columns as a table.

```
mtcars |> select(mpg, wt)
```

relocate() (data, ..., before = NULL, after = NULL) Move columns to new position.

```
mtcars |> relocate(mpg, cyl, after = last_col())
```

Use these helpers with select() and across()

```
e.g. mtcars |> select(mpg:cyl)
contains(match) num_range(prefix, range) 1: e.g., mpg:cyl
ends_with(match) all_of(v)any_of(x,...,vars) 1: e.g., gear
starts_with(match) matches(match)
```

MANIPULATE MULTIPLE VARIABLES AT ONCE

```
df <- tibble(x_1 = c(1, 2), x_2 = c(3, 4), y = c(4, 5))
```

across() (cols, funs, ..., names = NULL) Summarize or mutate multiple columns in the same way.

```
df |> summarize(across(everything(), mean))
```

c_across() (cols) Compute across columns in row-wise data.

```
df |> rowwise() |> mutate(x_total = sum(c_across(1:2)))
```

MAKE NEW VARIABLES

Apply **vectorized functions** to columns. Vectorized functions take

[Lien vers la cheatsheet](#)

Exercice 2 – les fonctions dplyr

► mutate()

- Permet d'ajouter une nouvelle colonne
 - Peut se baser sur le contenu d'une autre colonne
 - Par défaut, nouvelle colonne apparaît à l'extrémité droite de la banque de données
- Permet de modifier une colonne existante

```
> tbl_fictive
# A tibble: 3 x 3
  date_complete lettre chiffre
<chr>          <chr>   <dbl>
1 2021-01-01    a         1
2 2022-02-02    b         2
3 NA            c         3
```

```
> tbl_fictive %>%
+ mutate(chiffre_2 = chiffre*2,
+        lettre = toupper(lettre))
# A tibble: 3 x 4
  date_complete lettre chiffre chiffre_2
<chr>          <chr>   <dbl>   <dbl>
1 2021-01-01    A         1         2
2 2022-02-02    B         2         4
3 NA            C         3         6
```

Exercice 2 – les fonctions dplyr

► group_by() et summarise()

- Permettent d'effectuer un sommaire statistique d'une colonne(s) par groupe d'observation/ligne
- group_by() doit être utilisé avec d'autres fonctions de dplyr

```
> tbl_fictive_2
# A tibble: 12 x 3
  date_complete lettre chiffre
  <chr>         <chr>   <dbl>
1 2021-01-01    a         1
2 2021-01-01    b         2
3 2021-01-01    c         3
4 2021-01-01    a         1
5 2021-01-01    b         2
6 2021-01-01    c         3
7 2021-01-01    a         1
8 2021-01-01    b         2
9 2021-01-01    c         3
10 2021-01-01   a         1
11 2021-01-01   b         2
12 2022-02-02   c         3
```

Regroupement avec une variable

```
> tbl_fictive_2 %>%
+ group_by(lettre) %>%
+ summarise(n = n(),
+           moyenne = mean(chiffre))
# A tibble: 3 x 3
  lettre     n moyenne
  <chr> <int> <dbl>
1 a         4         1
2 b         4         2
3 c         4         3
```

Regroupement avec deux variables

```
> tbl_fictive_2 %>%
+ group_by(lettre, date_complete) %>%
+ summarise(n = n(),
+           mediane = median(chiffre))
`summarise()` has grouped output by 'lettre'.
# A tibble: 4 x 4
# Groups:   lettre [3]
  lettre date_complete     n mediane
  <chr>   <chr>         <int> <dbl>
1 a     2021-01-01         4         1
2 b     2021-01-01         4         2
3 c     2021-01-01         3         3
4 c     2022-02-02         1         3
```

Exercice 2 – les dates

► Les objets "date" et "date-time" sont stockés en nombre; représentent le nombre d'unités de temps depuis 1970-01-01

► Formats

- help(strptime) pour voir tous les formats possibles
 - %m = mois (1-12)
 - %B = nom du mois (Janvier-Décembre)
 - %U = semaine CDC (01-53)
- Ex. "%Y-%m-%d" = aaaa-mm-jj

► Fonction as.Date() du package base

```
> tbl_fictive_2 %>%
+ mutate(date_complete = as.Date(date_complete))
# A tibble: 12 x 3
  date_complete lettre chiffre
  <date>         <chr>   <dbl>
1 2021-01-01    a         1
2 2021-01-01    b         2
3 2021-01-01    c         3
```

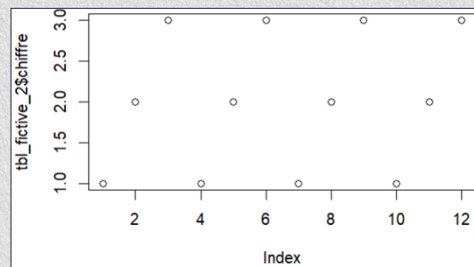

Exercice 2 – les fonctions lubridate

- ▶ Utiles pour le travail avec les objets "date" et "date-time"
- ▶ [Cheatsheet](#)
- ▶ Suit la convention ISO : jour de la semaine numérotés 1 à 7 (lundi = 1)
- ▶ Fonction `floor_date()`
 - Arrondit une date à la borne inférieure de l'unité de temps spécifié
 - Ex. obtenir la date de la semaine CDC correspondant au 2023-11-29

```
> date_atelier <- as.Date("2023-11-29")
> date_atelier
[1] "2023-11-29"
>
> dt_debut_sem_cdc <- floor_date(date_atelier,
+                               unit = "week",
+                               week_start = 7)
> dt_debut_sem_cdc
[1] "2023-11-26"
```

Exercice 2 – les graphiques de base

- ▶ Les fonctions du package `graphics` sont utiles pour visualiser les données
 - Ex. `boxplot()`, `plot()`, `hist()`
- ▶ Il faut nécessairement les utiliser avec le dialecte "R de base" et non "tidy"
 - Ex. `> plot(tbl_fictive_2$Schiffre)`
- ▶ Des options de mise en forme sont disponibles



Questions



Exercice 2 : Traitement et exportation de données



Exercice 2

- ▶ Travail avec les animateurs
- ▶ Retour sur l'exercice 2
 - Enjeux/bons coups au niveau du fonctionnement
 - Concepts à clarifier
 - Éléments intéressants (ex. échanges, solutions alternatives) à partager à l'ensemble du groupe

Présentation 3 : Visualisation des données



Exercice 3 – un aperçu

- ▶ Création et mise en forme de graphiques
 - Créer différents types de graphiques du nb de cas par semaine cdc et par sexe, Montérégie
 - **Exercice 3.1** : Créer un histogramme empilé du nb de cas par semaine cdc et par rts_infocentre avec la bd "sga"
 - Modifier les couleurs de l'histogramme créé à l'exercice 3.1
 - Créer, mettre en forme et exporter l'histogramme du nb de cas par semaine cdc et par sexe, Montérégie
 - Créer, mettre en forme et exporter le graphique à barres du nombre de cas par semaine CDC, ensemble du Québec
 - *Pour aller plus loin* : ajouter une courbe secondaire au graphique du nombre de cas par semaine CDC, ensemble du Québec

Exercice 3 – un aperçu

- ▶ Création et mise en forme de tableaux
 - Créer, mettre en forme et exporter un tableau qui illustre le nombre et la proportion des cas de sga selon le groupe d'âge en Montérégie
 - **Exercice 3.2** : Créer un tableau similaire à sga_age, mais avec la variable hospitalisation
 - *Pour aller plus loin* : créer un tableau incluant les données d'âge et d'hospitalisation et le mettre en forme
 - **Exercice 3.3** : En groupe, réfléchir à comment réunir les tableaux

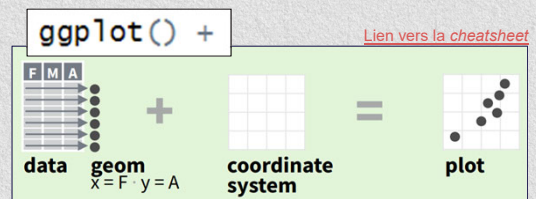
Exercice 3 – quelques fonctions

Package	Fonction	Utilité
ggplot2	ggplot()	Créer un graphique "ggplot" à partir d'éléments/blocs de code
ggplot2	ggsave()	Enregistrer une image d'un graphique "ggplot"
gt	gt()	Créer un tableau "gt" à partir d'une table de données
gt	gtsave()	Enregistrer un tableau "gt" en un fichier HTML
webshot2	paramètres additionnels dans la fonction gtsave()	Prend une capture d'écran à partir d'un lien URL. Dans l'atelier, on l'utilise pour créer et enregistrer une image .png à partir d'un tableau "gt" dont le format automatique est .html

Exercice 3 – les fonctions ggplot2

- ▶ Utiles pour créer un graphique à partir d'éléments/blocs de code à combiner à l'aide de l'opérateur « + »
- ▶ Chaque graphique commence par la fonction **ggplot()** et contient les éléments minimaux suivants :
 1. Banque de données
 2. geom : objet géométrique (courbe, histogramme, nuage de points, etc.)
 3. aesthetic mapping (aes) : système de coordonnées/valeurs des axes et autres éléments esthétiques

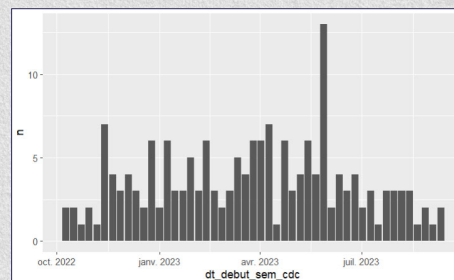
▶ [Cheatsheet](#)



Exercice 3 – la banque de données

- Options pour spécifier la bd à utiliser :
 - Devant l'opérateur %>% de dplyr()
 - Dans la fonction ggplot()

```
barre_r16 <- sga_cdc_sexe %>%  
  ggplot(aes(x = dt_debut_sem_cdc, y = n)) +  
  geom_col()  
  
#OU  
barre_r16 <-  
  ggplot(data = sga_cdc_sexe,  
        aes(x = dt_debut_sem_cdc, y = n)) +  
  geom_col()
```



Exercice 3 – la fonction aes()

- Options pour spécifier les valeurs x et y à l'intérieur de la fonction aes() :

```
barre_r16 <- sga_cdc_sexe %>%  
  ggplot() +  
  geom_col(aes(x = dt_debut_sem_cdc, y = n))  
  
#OU  
barre_r16 <- sga_cdc_sexe %>%  
  ggplot(aes(x = dt_debut_sem_cdc, y = n)) +  
  geom_col()
```

geom(aes(...))

- Les aes(x, y) sont utilisés par ce geom_col uniquement
- D'autres geom_() pourraient être ajoutés et contenir différents aes(x, y)

ggplot(aes(...))

- Les aes(x, y) seront utilisés/hérités par l'ensemble des geom_() si aucun autre aes(x, y) n'est spécifié

Exercice 3 – les fonctions geom_()

► Quelques exemples :

Fonction	Type de courbe	Nombre de variables représentées	Détail
geom_line()	Courbe	Deux variables	Les valeurs de X et de Y sont spécifiées dans l'aes()
geom_col()	Graphique à barres	Deux variables	Les valeurs de X et de Y sont spécifiées dans l'aes()
geom_bar()	Graphique à barres	Une variable	Seules les valeurs de X sont spécifiées dans l'aes(); Y correspond au nombre d'observations pour chaque X
geom_histogram()	Histogramme	Une variable <i>continue</i>	Seules les valeurs de X sont spécifiées dans l'aes(); Y correspond au nombre d'observations pour chaque intervalle/bâtonnets/bin de X



Noun project



Noun project



Noun project



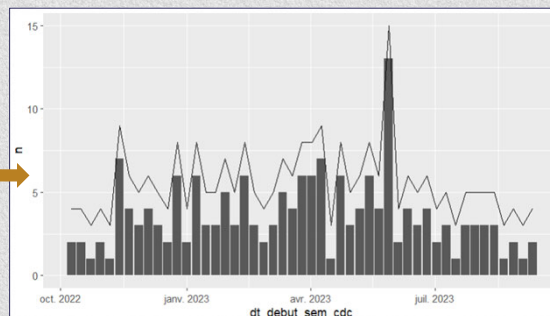
Noun project

Exercice 3 – les fonctions geom_()

► Un graphique peut contenir plusieurs geom_()

```
# aes(x,y) dans ggplot() hérités par tous les geom
barre_r16 <- sga_cdc_sexe %>%
  ggplot(aes(x = dt_debut_sem_cdc, y = n)) +
  geom_col() +
  geom_line()

# aes(x,y) dans ggplot() hérités par un seul geom
barre_r16 <- sga_cdc_sexe %>%
  ggplot(aes(x = dt_debut_sem_cdc, y = n)) +
  geom_col() +
  geom_line(aes(x = dt_debut_sem_cdc, y = n+2))
```



Exercice 3 – les paramètres des geom_()

- ▶ En plus des paramètres de base, soit la banque de données et l'aes() (hérités ou spécifiés), d'autres paramètres peuvent être modifiés à l'intérieur des fonctions geom_()
- ▶ Ex. avec geom_histogram() :

Paramètre	Utilité	Valeurs par défaut
stat	Transforme et illustre des variables de la bd	stat = "bin", effectue la somme du nombre d'observations par intervalle/bin de X
position	Ajuste la position des groupes	position = "stack", indique d'empiler les groupes
binwidth	Ajuste la largeur de l'intervalle/bin de X	binwidth = NULL, indique que la largeur de l'intervalle est déterminée automatiquement

Exercice 3 – d'autres composantes d'un graphique "ggplot"

Composante	Utilité	Exemple de fonction/paramètre
<i>Annotation</i>	Ajoute une annotation et est indépendant des aes()	geom_hline() geom_text()
<i>Aesthetic attributes/ Aesthetics (aes())</i>	Indique les propriétés visuelles des geom_() : x, y, couleurs, formes, <i>grouping</i>	type = "dotted", ligne pointillée shape = "square", point en forme de carré color = "blue", contour bleu
<i>Scales</i>	Ajoute et/ou ajuste des éléments de graphique (ex. titres, axes) et modifie les propriétés visuelles des geom_()	labs() expand_limits() scale_x_date() scale_colour_manual()
<i>Guides</i>	Ajoute et/ou ajuste les axes et la légende du graphique	sec_axis()
<i>Themes</i>	Ajuste la forme des éléments de graphiques qui ne sont pas des données	theme(axis.text = element_text(size = "5"))

Exercice 3 – l'utilisation des *aesthetics*

- ▶ *aesthetics* : ce que l'on voit (x, y, couleurs, formes, etc.)
- ▶ Ex : fill, soit la couleur de remplissage
 - Options pour référer à une couleur : "nom"/ "#code RGB"
 - Options pour spécifier le fill
 - À l'intérieur d'un aes() de la fonction geom_()/ggplot()
 - geom_histogram(aes(fill = sexe)) : le remplissage reflète les niveaux de la variable sexe
 - À l'extérieur d'un aes() de la fonction geom_()/ggplot()
 - geom_histogram(fill = "royalblue") : le remplissage est bleu

Exercice 3 – l'utilisation des *aesthetics*

```
# La couleur de remplissage représente les niveaux
## de la variable sexe (variable dans la bd)

barre_r16 <-

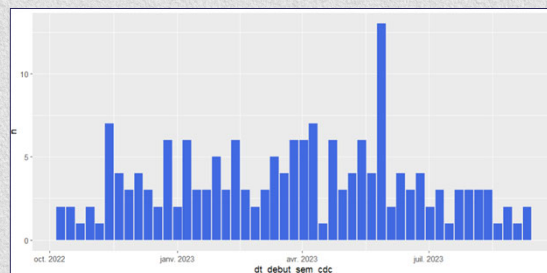
ggplot(data = sga_cdc_sexe,
       aes(x = dt_debut_sem_cdc, y = n, fill = sexe)) +
geom_col()
```



```
# La couleur de remplissage est bleu royal

barre_r16 <-

ggplot(data = sga_cdc_sexe,
       aes(x = dt_debut_sem_cdc, y = n)) +
geom_col(fill = "royalblue")
```

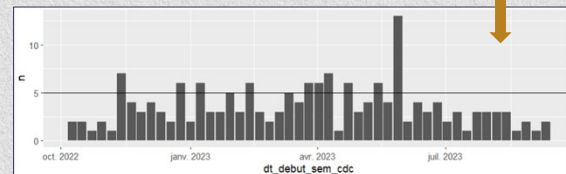


Exercice 3 – les objets "ggplot"

- ▶ Un graphique peut être stocké dans un objet "ggplot"
 - Pour l'afficher dans l'onglet *Plots*, exécuter le nom de l'objet
- ▶ Il est possible d'ajouter des éléments à un objet ggplot existant
- ▶ Cette logique s'applique aussi aux objets "gt"

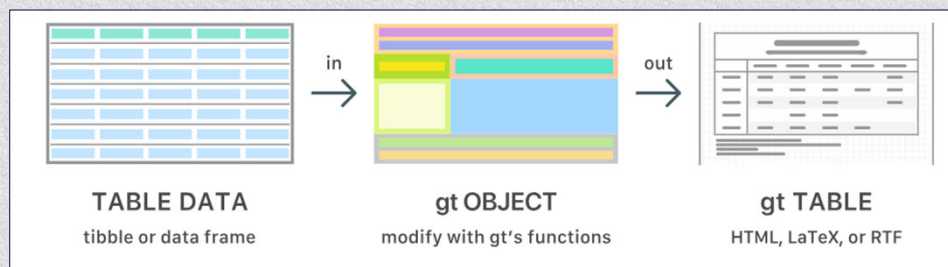
```
barre_r16 <-  
  ggplot(data = sga_cdc_sexe,  
         aes(x = dt_debut_sem_cdc, y = n)) +  
  geom_col()  
# Visualiser le graphique  
barre_r16
```

```
# Ajout d'une ligne horizontale à y=5  
barre_r16 +  
  geom_hline(yintercept = 5)  
barre_r16
```



Exercice 3 – les fonctions gt

- ▶ [Top 7 Packages for Making Beautiful Tables in R](#)
- ▶ Utiles pour mettre en forme des tableaux à partir d'éléments/blocs de code à combiner à l'aide de l'opérateur « %>% »
- ▶ Pour créer un tableau "gt", on commence par appliquer la fonction **gt()** à un tableau créé au préalable
- ▶ Les autres éléments servent à la mise en forme et à l'enregistrement

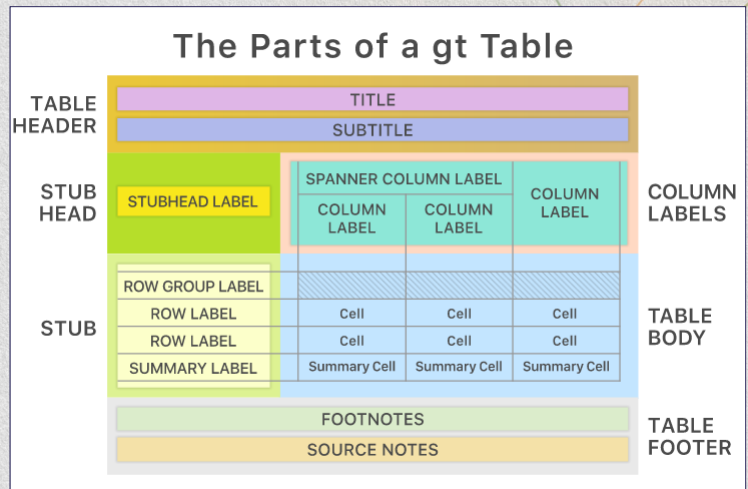


[Lien vers la page web](#)

Exercice 3 – les fonctions gt

► Ex.

- `tab_header()`
- `cols_label()`
- `cols_align()`
- `tab_footnote()`
- `tab_style()`



[Lien vers la page web](#)

Exercice 3 – les fonctions gt

► Exemple de base :

```
gt_tab <- tbl_fictive %>%  
  
  gt() %>%  
  
  tab_header(  
    title = "Tableau fictif, atelier JASP") %>%  
  
  tab_source_note(  
    "Source : données fictives, 2023-11-29")  
  
gt_tab
```

Tableau fictif, atelier JASP		
date_complete	lettre	chiffre
2021-01-01	a	1
2022-02-02	b	2
2023-03-03	c	3
Source : données fictives, 2023-11-29		

Affiché dans l'onglet Viewer

Exercice 3 – quelques fonctions gt de formatage

- ▶ `tab_style()` modifie le format de cellule(s) du tableau. Contient 3 paramètres :
 1. objet `gt*`
 2. `style` : ex. `cell_text()`
 3. `location` : ex. `cells_title()`
- ▶ `tab_options()` modifie le format des composantes du tableau. Contient *beaucoup* de paramètres :
 1. objet `gt*`
 2. paramètre : ex. `table.background.color =` , `table.font.size =`

**Spécifié au début du code du tableau*

Exercice 3 – les fonctions gt, exemple

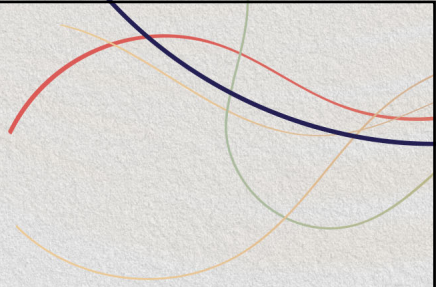
- ▶ Exemple incluant quelques fonctions de formatage :

```
gt_tab_format <- gt_tab %>%  
  
# Modifie noms colonnes  
cols_label(  
  date_complete = "Date",  
  lettre = "Lettre",  
  chiffre = "n") %>%  
  
# Modifie alignement colonnes  
cols_align(align = "center") %>%  
  
# Change le format du texte (gras) dans  
## le titre et les en-têtes de colonnes  
tab_style(cell_text(weight = "bold"),  
  locations = list(cells_title(),  
                    cells_column_labels())) %>%  
  
# Change certaines composantes du graphique  
tab_options(column_labels.background.color = "darkblue",  
  table.border.top.style = "hidden",  
  table.border.bottom.style = "hidden",  
  table.width = px(400))  
  
gt_tab_format
```

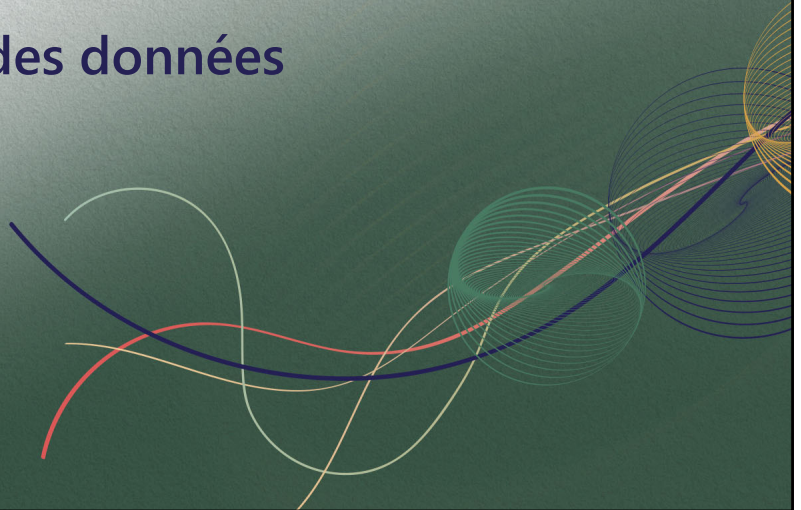
Date	Lettre	n
2021-01-01	a	1
2022-02-02	b	2
2023-03-03	c	3

Source : données fictives, 2023-11-29

Questions



Exercice 3 : Visualisation des données



Exercice 3

- ▶ Travail avec les animateurs
- ▶ Retour sur l'exercice 3
 - Concepts à clarifier
 - Éléments (ex. échanges, solutions alternatives) intéressants à partager à l'ensemble du groupe

Présentation 4 : Mise à jour efficace des données



Exercice 4 – un aperçu

- ▶ Importer les données brutes et les nettoyer en un seul morceau
- ▶ Déterminer la période analysée et la date d'extraction
- ▶ Créer un histogramme du nombre de cas par semaine CDC qui inclut des étiquettes de données - liste de cas
 - Insérer les objets date dans le code de l'histogramme
 - Déterminer et extraire le y max par "barre/bin" pour assurer que les étiquettes de données ne soient pas coupées
- ▶ Créer un graphique à barres du nombre de cas par semaine CDC - indicateur agrégé
 - Insérer les objets date dans le code du graphique à barres

Exercice 4 – un aperçu

- ▶ **Exercice 4.1** : Modifier le code du tableau des hospitalisations pour que les dates soient mises à jour automatiquement
 - Ajouter le filtre pour déterminer la période analysée
 - Insérer les objets date dans le titre et la note du tableau
- ▶ Exporter les figures en leur attribuant une date et une heure de production
- ▶ Actualiser les données au T2
 - **Exercice 4.2** :
 - Importer la liste de cas brute incluant une semaine additionnelle de données
 - Ajuster la valeur de `date_fin_der` et exécuter le code
 - *Pour aller plus loin : ajout de dates longues dans le titre d'une figure*

Exercice 4 – quelques fonctions

Package	Fonction	Utilité
base	paste() / paste0()	Permet de concaténer des termes (objet et/ou texte) avec un séparateur / sans séparateur
base	Sys.time()	Permet d'obtenir la date et l'heure actuelle
base	format()	Ajuster le format d'impression d'un objet

Exercice 4 – la mise à jour de données

- ▶ Ex. Analyse hebdomadaire : on ajoute une semaine CDC de données aux figures
 - Éléments à mettre à jour
 - La bd (et la plage de données)
 - Inclut le traitement de données
 - Les dates utiles aux analyses et à l'affichage dans la figure
 - La date à afficher dans le nom de la figure sauvegardée
 - Ces éléments peuvent être stockés dans des objets et insérés dans le code

Exercice 4 – la mise à jour de données

- ▶ La bd (et plage de données)
- ▶ Les dates de début de semaine CDC
 - **Début** de la 1re semaine CDC
 - **Début** de la dernière semaine CDC
- ▶ Les dates correspondant à la période analysée
 - **Début** de la 1re semaine CDC
 - **Fin** de la dernière semaine CDC
- ▶ Date d'extraction/mise à jour
- ▶ La date du système à afficher dans le nom de la figure sauvegardée

```
sga_qc_brut <-  
read_excel("00_bd_brut/nbcas_sga_ensqd_t2.xlsx",  
sheet = "Strept. A", # Indique la feuille à lire  
range = "A4:C56") # Indique la plage de cellules
```

À utiliser pour filtrer la variable de date de début de semaine CDC

À insérer dans le code de la figure (titres et spécification de l'axe des X)

À insérer dans le code de la figure (note sous le graphique/caption)

```
format(Sys.time(), "%Y-%m-%d_%H%M")
```

Exercice 4 – la mise à jour de données

- ▶ Bonnes pratiques
 - Placer les objets à initialiser en début de code (bd, dates)
 - Valider que les bd brutes sont bien importées, spécialement les indicateurs agrégés (plage de données peut changer)
 - Créer une routine de validation dans le code (à explorer dans une prochaine formation)

Questions



Exercice 4 :
Mise à jour efficace des données



Exercice 4

- ▶ Travail avec les animateurs
- ▶ Retour sur l'exercice 4
 - Concepts à clarifier
 - Éléments (ex. échanges, solutions alternatives) intéressants à partager à l'ensemble du groupe

Discussion 2

Discussion 2 : retour sur l'atelier

- ▶ Êtes-vous (toujours) intéressés à intégrer le logiciel R comme outil de vigie-surveillance? Pourquoi?
- ▶ Comment avez-vous trouvé l'atelier?
 - Contenu
 - Format

Conclusion

Ressources additionnelles

Sujet	Hyperliens
Cheatsheet R	https://posit.co/resources/cheatsheets/?type=posit-cheatsheets&_page=1/
Introduction	https://cran.r-project.org/doc/contrib/Paradis-rdebuts_fr.pdf https://bookdown.org/daniel_dauber_io/r4np_book/ https://r.qcbs.ca/fr/workshops/
Ouvrages de référence	https://r4ds.hadley.nz/ https://rstudio-education.github.io/hopr/index.html https://adv-r.hadley.nz/index.html
ggplot2	https://r-graphics.org/ (ouvrage de référence) https://beanumber.github.io/sds192/lab-ggplot2.html#Introduction (ouvrage de référence) http://www.sthda.com/english/wiki/ggplot2-histogram-plot-quick-start-guide-r-software-and-data-visualization (introduction) https://ggplot2.tidyverse.org/reference/ (composantes) https://ggplot2.tidyverse.org/articles/ggplot2-specs.html (spécifications de l'aes()) https://ggplot2.tidyverse.org/reference/aes_eval.html (transformation de variables illustrées) https://ggplot2.tidyverse.org/articles/faq-axes.html (FAQ sur les axes) https://ggplot2.tidyverse.org/reference/sec_axis.html (création d'un axe Y secondaire) https://r-graph-gallery.com/line-chart-dual-Y-axis-ggplot2.html (création d'un axe Y secondaire) https://stackoverflow.com/questions/51456307/how-to-add-a-legend-for-the-secondary-axis-ggplot (ajout légende pour axe Y secondaire) https://r-charts.com/ggplot2/titles/ (ajout et modification de titres, sources, etc.) https://isabella-b.com/blog/ggplot2-theme-elements-reference/ (définition et localisation des thèmes)

Ressources additionnelles

Sujet	Hyperliens
gt	https://gt.rstudio.com/articles/gt.html (introduction) https://gt.rstudio.com/ (introduction) https://gt.rstudio.com/reference/tab_options.html (modification du format des composantes du tableau) https://gt.rstudio.com/reference/tab_style.html (modification du format de cellule(s) du tableau) https://gt.rstudio.com/reference/gtsave.html + https://cran.r-project.org/web/packages/webshot2/webshot2.pdf (enregistrer un tableau gt en .png) https://www.danielsjoberg.com/gtsummary/ (package gtsummary)
Couleurs	https://www.nceas.ucsb.edu/sites/default/files/2020-04/colorPaletteCheatsheet.pdf https://sjmgarnier.github.io/viridis/reference/scale_viridis.html https://www.stat.ubc.ca/~jenny/STAT545A/block14_colors.html
Épidémiologie	https://epirhandbook.com/en/index.html (ouvrage de référence) https://www.r4epi.com/index.html (ouvrage de référence)

Remerciements

- ▶ Direction de santé publique de la Montérégie
- ▶ Personnes organisatrices et membres du comité scientifique des JASP
- ▶ Personnes formatrices et membres du comité scientifique de l'atelier
- ▶ Personnes participantes à l'atelier

Bibliographie – ordre d'apparition

- Perihara, D. (2023). *Graph* [image en ligne]. Noun Projet. <https://thenounproject.com/icon/graph-6253384/>
- Dsouza, T. (2017). *Data table* [image en ligne]. Noun Projet. <https://thenounproject.com/icon/data-table-869490/>
- Economic Theory Blog. (2017, 13 mai). *R – PROS AND CONS*. <https://economicstheoryblog.com/2017/05/13/r-pros-and-cons/>
- Harris, J. (2023, 20 septembre). *Advantages and Disadvantages of R programming*. The knowledge academy. <https://www.theknowledgeacademy.com/blog/advantages-and-disadvantages-of-r-programming/>
- Masic, I., Jankovic, S. M., et Begic, E. (2019). PhD Students and the Most Frequent Mistakes During Data Interpretation by Statistical Analysis Software. *Studies in health technology and informatics*, 262, 105–109. <https://doi.org/10.3233/SHT1190028>
- Data Fair. (s. d.). *Pros and Cons of Power BI – The Bright & the Dull side of visualization suite*. <https://data-flair.training/blogs/power-bi-advantages-and-disadvantages/>
- SoftwareOne. (2022, 22 septembre). *Power BI and R - Using R in Power BI - Predictive Analytics using Power BI*. <https://www.softwareone.com/en/blog/articles/2017/10/02/visualizing-data-r-script-power-bi>
- GeeksforGeeks. (2023, 5 juillet). *R Data Types*. <https://www.geeksforgeeks.org/r-data-types/>
- ETH Zürich. (s. d.). *R: Classes Corresponding to Basic Data Types*. <https://stat.ethz.ch/R-manual/R-devel/library/methods/html/BasicClasses.html>
- Paradis, E. (2005). *R pour les débutants*. Institut des Sciences de l'Évolution, Université Montpellier II. https://cran.r-project.org/doc/contrib/Paradis-rdebuts_fr.pdf
- Kosourova, E. (2022, 15 Juin). *How to Write Functions in R (with 18 Code Examples)*. Dataquest. <https://www.dataquest.io/blog/write-functions-in-r/>
- RDocumentation. (s. d.). *mean: Arithmetic Mean*. <https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/mean>

Bibliographie – ordre d'apparition

- Datacamp. (2019, mars). *R Packages: A Beginner's Tutorial*. <https://www.datacamp.com/tutorial/r-packages-guide>
- Wikipedia. (2023, 16 novembre). *ISO week date*. https://en.wikipedia.org/wiki/ISO_week_date
- Wickham, H., Spinu, V. et Grolemund, G. (s. d.). *Get/set weeks component of a date-time*. Posit. <https://lubridate.tidyverse.org/reference/week.html>
- Unicode. (s. d.). *Home*. <https://home.unicode.org/>
- Python Software Foundation. (2023, 20 novembre). *codecs — Codec registry and base classes*. <https://docs.python.org/3/library/codecs.html>
- Wickham, H., François, R., Henry, L., Müller, K. et Vaughan, D. (s. d.). *Introduction to dplyr*. Posit. <https://dplyr.tidyverse.org/articles/dplyr.html>
- Posit. (2023). *Posit Cheatsheets*. <https://posit.co/resources/cheatsheets/?type=posit-cheatsheets&page=1/>
- Wickham, H., François, R., Henry, L., Müller, K. et Vaughan, D. (s. d.). *Create, modify, and delete columns*. Posit. <https://dplyr.tidyverse.org/reference/mutate.html>
- Wickham, H., François, R., Henry, L., Müller, K. et Vaughan, D. (s. d.). *Summarise each group down to one row*. Posit. <https://dplyr.tidyverse.org/reference/summarise.html>
- Dai, Y. (2019, avril). *Dates and Times*. https://shanghai.hosting.nyu.edu/data/r/dates-and-times.html#dates_and_times_classes
- Yau, N. (2007). *Comparing ggplot2 and R Base Graphics*. FlowingData. <https://flowingdata.com/2016/03/22/comparing-ggplot2-and-r-base-graphics/>
- Wickham, H., Chang, W., Henry, L., Pedersen, T., Takahashi, K., Wilke, C., Woo, K., Yutani, H. et Dunnington, D. (s. d.). *Create a new ggplot*. Posit. <https://ggplot2.tidyverse.org/reference/ggplot.html>
- Posit. (2023, juillet). *Data visualization with ggplot2 : : CHEATSHEET*. <https://rstudio.github.io/cheatsheets/data-visualization.pdf>

Bibliographie – ordre d'apparition

- Wickham, H., Chang, W., Henry, L., Pedersen, T., Takahashi, K., Wilke, C., Woo, K., Yutani, H. et Dunnington, D. (s. d.). *Histograms and frequency polygons*. Posit. https://ggplot2.tidyverse.org/reference/geom_histogram.html
- Wickham, H., Chang, W., Henry, L., Pedersen, T., Takahashi, K., Wilke, C., Woo, K., Yutani, H. et Dunnington, D. (s. d.). *Bar charts*. Posit. https://ggplot2.tidyverse.org/reference/geom_bar.html
- Chang, W. (2018). *R Graphics Cookbook, 2nd edition*. O'Reilly. <https://r-graphics.org/recipe-bar-graph-basic-bar>
- Vectlab. (2019). *Bar chart* [image en ligne]. Noun Projet. <https://thenounproject.com/icon/bar-chart-2418974/>
- Gamma Designs. (2018). *Line chart* [image en ligne]. Noun Projet. <https://thenounproject.com/icon/line-chart-1960163/>
- Wickham, H., Chang, W., Henry, L., Pedersen, T., Takahashi, K., Wilke, C., Woo, K., Yutani, H. et Dunnington, D. (s. d.). *Function reference*. Posit. <https://ggplot2.tidyverse.org/reference/>
- Wickham, H., Chang, W., Henry, L., Pedersen, T., Takahashi, K., Wilke, C., Woo, K., Yutani, H. et Dunnington, D. (s. d.). *Create a new layer*. Posit. <https://ggplot2.tidyverse.org/reference/layer.html>
- Wickham, H., Chang, W., Henry, L., Pedersen, T., Takahashi, K., Wilke, C., Woo, K., Yutani, H. et Dunnington, D. (s. d.). *Aesthetic specifications*. Posit. <https://ggplot2.tidyverse.org/articles/ggplot2-specs.html>
- Baumer, B., et Crouser, J. R. (s. d.). *Aesthetic Mapping vs. Assignment*. https://beanumber.github.io/sds192/lab-ggplot2.html#Aesthetic_Mapping_vs_Assignment
- Bryan, J. (s. d.). *Using colors in R*. UBC Department of Statistics. https://www.stat.ubc.ca/~jenny/STAT545A/block14_colors.html
- Frazier, M. (s. d.). *R Color Palette Cheat Sheet*. National Center for Ecological Analysis and Synthesis. <https://www.nceas.ucsb.edu/sites/default/files/2020-04/colorPaletteCheatsheet.pdf>

Bibliographie – ordre d'apparition

Madhugiri, D. (2022, 7 octobre). *Top 7 Packages for Making Beautiful Tables in R*. Towards Data Science. <https://towardsdatascience.com/top-7-packages-for-making-beautiful-tables-in-r-7683d054e541>

Iannone, R., Cheng, J., Schloerke, B., Hughes, E., Lauer, A., et Seo, J. (s. d.). *Easily Create Presentation-Ready Display Tables*. Posit. <https://gt.rstudio.com/>

Iannone, R., Cheng, J., Schloerke, B., Hughes, E., Lauer, A., et Seo, J. (s. d.). *Introduction to Creating gt Tables*. Posit. <https://gt.rstudio.com/articles/gt.html>

Iannone, R., Cheng, J., Schloerke, B., Hughes, E., Lauer, A., et Seo, J. (s. d.). *Add custom styles to one or more cells*. Posit. https://gt.rstudio.com/reference/tab_style.html

Iannone, R., Cheng, J., Schloerke, B., Hughes, E., Lauer, A., et Seo, J. (s. d.). *Modify the table output options*. Posit. https://gt.rstudio.com/reference/tab_options.html

Iannone, R., Cheng, J., Schloerke, B., Hughes, E., Lauer, A., et Seo, J. (s. d.). *Helper for defining custom borders for table cells*. Posit. https://gt.rstudio.com/reference/cell_borders.html

Merci!