

# Document partagé "Introduction au logiciel R - 2025"

## Mercredi matin : rappels

Installation du package du cours :

```
remotes::install_github("vguillemot/intro2r")
```

```
> install.packages("remotes")
```

```
trying URL
```

```
'https://cran.rstudio.com/bin/macosx/big-sur-arm64/contrib/4.4/remotes_2.5.0.tgz'
```

```
Content type 'application/x-gzip' length 430216 bytes (420 KB)
```

```
=====
```

```
downloaded 420 KB
```

```
install.packages(remotes)
```

```
Error in install.packages : objet 'remotes' introuvable
```

```
→ explication il manquait les guillemets dans la commande
```

```
Error message:
```

```
Skipping install of 'intro2r' from a github remote, the SHA1 (3cdb1c0b) has not changed since last install.
```

```
Use `force = TRUE` to force installation
```

```
→ le package est déjà installé
```

```
> install.packages("intro2r")
```

```
Warning in install.packages :
```

```
package 'intro2r' is not available for this version of R
```

```
A version of this package for your version of R might be available elsewhere,
```

```
see the ideas at
```

```
https://cran.r-project.org/doc/manuals/r-patched/R-admin.html#Installing-packages
```

```
→ il faut utiliser la fonction remotes::install_github parce que intro2r n'est pas un package CRAN
```

```
> remotes::install_github("vguillemot/intro2r")
```

```
Downloading GitHub repo vguillemot/intro2r@HEAD
```

```
These packages have more recent versions available.
```

```
It is recommended to update all of them.
```

```
Which would you like to update?
```

```
1: All
2: CRAN packages only
3: None
4: rlang      (1.1.3 -> 1.1.5 ) [CRAN]
5: glue       (1.7.0 -> 1.8.0 ) [CRAN]
6: cli        (3.6.2 -> 3.6.4 ) [CRAN]
7: withr      (3.0.0 -> 3.0.2 ) [CRAN]
8: pillar     (1.9.0 -> 1.10.1) [CRAN]
9: colorspace (2.1-0 -> 2.1-1 ) [CRAN]
10: R6         (2.5.1 -> 2.6.1 ) [CRAN]
11: gtable     (0.3.5 -> 0.3.6 ) [CRAN]
```

Enter one or more numbers, or an empty line to skip updates:

```
library(remotes)
```

Enter one or more numbers, or an empty line to skip updates:

```
remotes::install_github("vguillemot/intro2r")
```

Enter one or more numbers, or an empty line to skip updates:

```
remotes::install_github("vguillemot/intro2r")
```

Enter one or more numbers, or an empty line to skip updates:

```
library(intro2r)
```

Enter one or more numbers, or an empty line to skip updates:

```
Running `R CMD build`...
```

```
* checking for file
```

```
'C:\Users\Libre\AppData\Local\Temp\RtmpGMQ1w6\remotes10f42f526b0c\
vguillemot-intro2r-3cdb1c0/DESCRIPTION' ... OK
```

```
* preparing 'intro2r':
```

```
* checking DESCRIPTION meta-information ... OK
```

```
* checking for LF line-endings in source and make files and shell
scripts
```

```
* checking for empty or unneeded directories
```

```
* building 'intro2r_1.0.2025.tar.gz'
```

```
Installation du package dans
```

```
'C:/Users/Libre/AppData/Local/R/win-library/4.3'
```

```
(car 'lib' n'est pas spécifié)
```

```
* installing *source* package 'intro2r' ...
```

```
** using staged installation
```

```
** R
```

```
** data
```

```
*** moving datasets to lazyload DB
```

```
** inst
```

```
** byte-compile and prepare package for lazy loading
```

```
Messages d'avis :
```

```
1: le package 'ggplot2' a été compilé avec la version R 4.3.3
```

```
2: le package 'dplyr' a été compilé avec la version R 4.3.3
```

```
** help
```

```
*** installing help indices
```

```
** building package indices
```

```
** installing vignettes
```

```

** testing if installed package can be loaded from temporary
location
Avis : le package 'ggplot2' a été compilé avec la version R 4.3.3
Avis : le package 'dplyr' a été compilé avec la version R 4.3.3
** testing if installed package can be loaded from final location
Avis : le package 'ggplot2' a été compilé avec la version R 4.3.3
Avis : le package 'dplyr' a été compilé avec la version R 4.3.3
** testing if installed package keeps a record of temporary
installation path
* DONE (intro2r)
→ tout va bien mais le dialogue ne s'est pas forcément bien passé

```

```

> remotes::install_github("vguillemot/intro2r")
Error: object 'remotes' not found
→ il manque un "." dans remotes::install_github("vguillemot/intro2r")

```

## Avec les fonctions du package pak

```

pak::pak("vguillemot/intro2r")

```

## Exercice sur les booléens

Soit le vecteur aléatoire suivant :

```

poids <- rnorm(100, mean = 70, sd = 10)

```

- 1) Je veux un vecteur de booléens qui est vrai quand le poids est compris entre 65 et 75.

Réponse :

```

poids_entre_65_75 <- (poids >= 65) & (poids <= 75)
poids_entre_65_75

```

Ou bien :

```

poids_65_75 <- (poids <= 75 & poids >= 65)

```

**Erreurs :**

```

poids_entre_65_75 <- (df$weight_kg >= 65) & (df$weight_kg <= 75)
→ Attention erreur : df n'existe pas

```

```

Error message: Error: unexpected '<' in "poids_entre_65_75 <-
(poids >=65) & (poids=<"
> poids_entre_65_75

```

```

Error: object 'poids_entre_65_75' not found

```

→ l'opérateur "<=" n'existe pas, la bonne syntaxe est "<="

- 2) Maintenant je veux les **valeurs** de poids comprises entre 65 et 75

Méthode 1:

```

poids[poids <= 75 & poids >= 65]

```

Création dataframe:

```
dat <- data.frame(poids)
```

## Exercice : chargement et exploration des données “fruits et légumes”

### 1. Chargement des données

## Première possibilité : fonctionne parce que ce sont des données associées au package intro2r

```
data("fruveg", package = "intro2r")
```

## Deuxième possibilité (utile pour les données personnelles)

```
library(readxl)
```

```
fruveg <- read_excel("fruits_and_veggies.xlsx")
```

### 2. extraire les fruits et légumes dont la teneur en eau est supérieure à 90%

```
eau_90 <- fruveg[fruveg$water > 90, ]
```

### 3. extraire les fruits crus

```
fruits_crus <- fruveg[fruveg$group == "raw fruit",]
```

### 4. extraire les légumes cuits contenant moins de 1g/100g de sucres

```
legumes_cuits_peu_sucres <- fruveg[fruveg$group=="cooked vegetable" & fruveg$sugar < 1,]
```

## Erreurs et débogage

```
> fruit_veggies[fruit_veggies$water>=90, ]
```

```
Error in base::nchar(wide_chars$test, type = "width") :
```

```
lazy-load database
```

```
'/Library/Frameworks/R.framework/Versions/4.4-x86_64/Resources/lib  
rary/cli/R/sysdata.rdb' is corrupt
```

```
In addition: Warning messages:
```

```
1: In base::nchar(wide_chars$test, type = "width") :
```

```
restarting interrupted promise evaluation
```

```
2: In base::nchar(wide_chars$test, type = "width") :
```

```
internal error -3 in R_decompress1
```

→ BDD de package mal chargée donc il faut redémarrer Rstudio

```
Error in type.convert.default(data[[i]], as.is = as.is[i], dec =  
dec, :
```

```
invalid multibyte string at '<b6><85>m<9a><87>8F<82>5<
```

```
L`<d7>□-<de>`<ad>A<8c><85>2{dp<a9><f9>9R'
```

```
In addition: Warning messages:
```

```
1: In read.table("/Users/melan/Downloads/nutrimenu.xlsx", header =  
TRUE, :
```

```
line 1 appears to contain embedded nulls
```

```
2: In read.table("/Users/melan/Downloads/nutrimenu.xlsx", header =  
TRUE, :
```

```

    line 2 appears to contain embedded nulls
3: In read.table("/Users/melan/Downloads/nutrimenu.xlsx", header =
TRUE,  :
    line 3 appears to contain embedded nulls
4: In scan(file = file, what = what, sep = sep, quote = quote, dec
= dec,  :
    embedded nul(s) found in input

```

→ Utilisation de read.table sur un fichier Excel, il faut utiliser autre chose

```

> raw_fruit <- fruits_and_veggies[fruits_and_veggies$type == "raw fruit", ]
Error in `fruits_and_veggies[fruits_and_veggies$type == "raw fruit", ]`:
! Can't subset rows with `fruits_and_veggies$type == "raw fruit"`.
✖ Logical subscript `fruits_and_veggies$type == "raw fruit"` must be size 1
or 33, not 0.
Run `rlang::last_trace()` to see where the error occurred.
Warning message:
Unknown or uninitialised column: `type`.

```

→ type n'existe pas dans la base de données

```

fruits_and_veggies <- read_excel("fruits_and_veggies.xlsx")
is.data.frame(fruits_and_veggies)
fv<-fruits_and_veggies
is.data.frame(fv)

```

#EXTRAIRE F&L teneur>90%

```
teneursup_90<-fv$water[fv$water>90]
```

```
teneur_eau_90 <- fruveg$water[fruveg>90]
```

```

> crus<-fruveg[fruveg$group,raw]
Error in `fruveg[fruveg$group, raw]`:
! Can't subset columns with `raw`.
✖ `raw` must be logical, numeric, or character, not a function.
Run `rlang::last_trace()` to see where the error occurred.
Error in `fruveg[fruveg$group == "raw fruit"]`:
! Can't subset columns with `fruveg$group == "raw fruit"`.
✖ Logical subscript `fruveg$group == "raw fruit"` must be size 1 or 19, not
33.
Run `rlang::last_trace()` to see where the error occurred.

```

→ Il manque la virgule

```

> cooked_veggies1 <- fruits_and_veggies$group == "raw
fruit"[fruits_and_veggies$group$sugar == "cooked vegetables", ]
Error in fruits_and_veggies$group$sugar :
  $ operator is invalid for atomic vectors

```

→ on ne peut pas mettre 2 \$

```
fv[fv$group=="cooked vegetables" & "df$sugar"<1,]
```

→ Double problème : les guillemets et df

```
fv[fv$group=="cooked vegetables" & fv$sugar<1,]  
# A tibble: 0 × 19  
# i 19 variables: name <chr>, group <chr>, energy <dbl>, water <dbl>,  
proteins <dbl>,  
#   carbohydrates <dbl>, lipids <dbl>, sugar <dbl>, fibers <dbl>, alcohol  
<dbl>,  
#   calcium <dbl>, copper <dbl>, iron <dbl>, magnesium <dbl>, manganese  
<dbl>,  
#   phosphorus <dbl>, potassium <dbl>, zinc <dbl>, vitaminC <dbl>  
→ Il y a un s en trop à vegetables
```

# Mercredi Après-midi : Visualisation avec ggplot2

Pour commencer :

```
library(ggplot2)
data("fruveg", package = "intro2r")
fruveg$group
table(fruveg$group)
# Pour obtenir un barplot :
barplot(table(fruveg$group))
```

```
## Exemple de ggplot
ggplot(fruveg,
       aes(x = sugar,
           y = energy,
           color = group)) +
  geom_point() +
  geom_smooth()
```

## Exercice : reproduire le graphe

```
fruveg$SucreSup5 <- fruveg$sugar > 5
```

Ou si on veut les appeler autrement que TRUE FALSE:

```
fruveg$SucreSup5 <- ifelse(fruveg$sugar > 5, "Sup 5", "Inf 5")
```

```
ggplot(fruveg,
       aes(x = SucreSup5,
           fill = SucreSup5)) +
  geom_bar()
```

##Avec geom\_bar

```
ggplot(fruveg, aes(group)) +
  geom_bar(fill = "steelblue")
```

## Avec geom\_col

```
vitamine <- data.frame(
  groupe = c("compote", "crus", "exotique", "secs"),
  VitamineC = c(14.2, 12.2, 32.1, 0.3))
```

```
ggplot(
  data = vitamine,
  mapping = aes(x = groupe, y = VitamineC)) +
  geom_col()
```

## Exercice : diagramme en bâtons de la teneur en eau

Comment modifier le code précédent pour faire un diagramme en bâtons montrant le nombre de fruits et légumes ayant une teneur en Eau supérieure à 80 g / 100 g par groupe ?

```
fruveg$water80 <- fruveg$water > 80
ggplot(fruveg, aes(x = water80, color = group, fill =
water80)) +
  geom_bar()
=> FAUX
```

### Proposition 1

```
eau_80 <- fruveg[fruveg$water > 80, ]

ggplot(
  data = eau_80,
  mapping = aes(x=group, fill = group)) +
  geom_bar()
```

### Autre solution :

```
ggplot(
  data = fruveg,
  mapping = aes(
    x = group,
    fill = water >= 80)) +
  geom_bar()
```

### 3ème solution :

```
library(dplyr)
#filter fruits avec water content > 80
filtered_data <- fruveg %>%
  filter(water > 80)
#Create a ggplot avec le numero des fruits/group
ggplot(filtered_data, aes(x = group)) +
  geom_bar(fill = "steelblue") +
  labs(title = "Counts of fruits with water content > 80%",
x= "Fruit category"
y= "Number of fruits")

#Counting :-)
group_counts <- fruveg %>%
  filter(water>80) %>%
  group_by(group) %>%
  summarise(count =n())
print (group_counts)
```



```
library(dplyr)
fruveg %>%
  group_by(group) %>%
  count(water >= 80) %>%
  filter(`water >= 80`) %>%
  ggplot(aes(x = group, y = n )) +
  geom_col()
```

### Histogramme de la teneur en sucre

```
ggplot(fruveg, aes(sugar)) +
  geom_histogram(breaks = seq(0, 20, 2.5))
# avec plus de paramètres
ggplot(fruveg, aes(sugar)) +
  geom_histogram(breaks = seq(0, 20, 2.5),
                 fill = "steelblue",
                 color = "white")
```

### Diagramme de Tukey

```
ggplot(data=fruveg, aes(x=group, y=sugar)) +
  geom_boxplot(fill = "pink")
```

```
bimodal <- data.frame(
  x = c(
    rnorm(100, 2),
    rnorm(100, 20))

ggplot(bimodal, aes(x)) +
  geom_histogram()
```

### Exemple de données simulées bimodales

```
bimodal <- data.frame(
  x = c(
    rnorm(n = 100, mean = 2),
    rnorm(n = 100, mean = 20))

ggplot(bimodal, aes(x)) +
  geom_histogram()

ggplot(bimodal, aes(x = x)) +
  geom_violin(aes(y = "Violon"), trim = FALSE) +
  geom_boxplot(aes(y = "Tukey"))
```

## Exercice : reproduction de diagramme en violons

```
ggplot(fruveg,
      aes(x = fibers > 1.5,
          y = proteins,
          fill = ***)) +
  geom_***()

ggplot(fruveg,
      aes(x = fibers > 1.5,
          y = proteins,
          fill = fibers>1.5)) + geom_violin()
```

## Exercice : reproduire le thème

```
ggplot(fruveg, aes(fibers)) +
  geom_boxplot() +
  theme_***()

ggplot(fruveg, aes(y=fibers)) +
  geom_boxplot() +
  theme_classic()
```

## Exercice : reproduction du nuage de points

```
ggplot(fruveg,
      aes(x = sugar,
          y = proteins,
          *** = magnesium,
          *** = ***)) +
  geom_***() +
  ***(title = "Fruits et légumes",
      x = "Sucres (g/100 g)",
      y = "Protéines, N x 6.25 (g/100 g)",
      size = "Magnésium\n(mg/100 g)",
      ***= "Groupe") +
  theme_***()

nuage de point : proposition mélanie
# nuage de points
ggplot(fruveg,
      aes(x = sugar,
```

```

        y = proteins,
        size = magnesium,
        color = group)) +
geom_point() +
labs(title = "Fruits et légumes",
      x = "Sucres (g/100 g)",
      y = "Protéines, N x 6.25 (g/100 g)",
      size = "Magnésium\n(mg/100 g)",
      color= "Groupe") +
theme_minimal()

```

## Exercice : trouver les fonctions de modification d'échelle

```

ggplot(fruveg,
       aes(phosphorus,
           calcium)) +
geom_point(color = "white") +
scale_x_log10() +
scale_y_log10() +
labs(x = "log10(Phosphore)",
      y = "log10(Calcium)") +
theme_dark()

```

## Pour sauvegarder un graphe

```

g <- ggplot(fruveg, aes(group)) +
  geom_bar()
print(g)

ggsave(filename = "mongraphe.png", plot = g)

```

## Jeudi Matin : visualisation plus avancée

Pour installer le package RColorBrewer (si besoin), utiliser la commande suivante :

```
install.packages("RColorBrewer")
```

### Exercice : reproduire le diagramme en bâtons

```
# Solution 1
a <- (1:5)
barplot(c(a, a), col=c("#B4EEB4", "#EEB422"))

# Solution 2
barplot(c(1:5, 1:5), col=c("#104E8B", "#EE3B3B"))

# Solution 3
barplot(c(1, 2, 3, 4, 5, 1, 2, 3, 4, 5), col=c("#436EEE", "#FF4040"))

# Solution 4
barplot(rep(1:5, 2), col = c(4, 2))

## Pour afficher toutes les palettes de RColorBrewer
display.brewer.all()
## Pour choisir 3 couleurs de la palette "Set3"
brewer.pal(n = 3, name = "Set3")
```

### Exercice : reproduire le diagramme en bâtons avec brewer.pal

```
pal <- brewer.pal(***, ***)
barplot(rep(1, 7),
        col = pal,
        axes = ***,
        border = ***)

pal <- brewer.pal(n=7, name="Set1")
barplot(rep(1, 7),
        col = pal,
        axes = FALSE,
        border = NA)

# Modification des couleurs dans ggplot
ggplot(fruveg, aes(x = group,
                  fill = group)) +
  geom_bar() +
  scale_fill_brewer(palette = "Set1") +
  theme_bw()
```

```
ggplot(fruveg, aes(x = phosphorus,
                  y = calcium,
                  color = magnesium)) +
  geom_point() +
  scale_color_distiller(palette = "PuOr") +
  theme_bw()
```

## Cartes de chaleur

```
library(pheatmap)

colfun <- colorRampPalette(
  c("darkorchid",
    "white",
    "limegreen"))
fruvegDF <- data.frame(
  fruveg[, -c(1, 10)],
  row.names = make.unique(fruveg$name))
annotLignes <- fruvegDF[, "group",
  drop = FALSE]

pheatmap(
  fruvegDF[, -1],
  cluster_rows = FALSE,
  scale = "column",
  show_rownames = FALSE,
  cellwidth = 10,
  color = colfun(20),
  annotation_row = annotLignes
)
```

Question : pourquoi le nid d'abeille est au milieu est pas autour des points ? merci

```
ggplot(data=trial,
       aes(death, age, fill = death))+geom_violin()+geom_beeswarm()
theme_minmal()
```

# Jeudi après-midi : tests et analyses statistiques

## Le jeu de données "trial"

```
# Chargement des packages
library(gtsummary)
library(ggplot2)
library(ggbeeswarm)
library(performance)

# Le jeu de données trial
View(trial)

# Distribution de l'âge en fonction du stade de la tumeur
ggplot(trial, aes(stage, age, fill = stage)) +
  geom_violin(trim = FALSE) +
  geom_beeswarm()

# Régression logistique
res.log <- glm(response ~ trt + age + marker + stage, # modele
               data = trial, # data
               family = binomial()) # famille de modèle

# Utilisation de tbl_regression
plot(res.log)
performance::model_performance(res.log)

res.lm <- lm(energy ~ proteins + sugar + fibers + water,
            data = fruveg)
summary(res.lm)
tbl_regression(res.lm)

##### Autre exemple de régression logistique : les données
"diabète" #####
# lien :
https://www.sthda.com/english/articles/36-classification-methods-essentials/151-logistic-regression-essentials-in-r/

## Chargement des packages
library(ggplot2)
library(gtsummary)
library(performance)
```

```

## Thème ggplot
theme_set(theme_bw())

## Les données de diabète
data("PimaIndiansDiabetes2", package = "mlbench")
dat.nona <- na.omit(PimaIndiansDiabetes2)
dat.nona$random <- rnorm(nrow(dat.nona))
## Modèle de régression logistique
res.log <- glm(diabetes ~., data = dat.nona, family = binomial)
res.log <- glm(diabetes ~ random, data = dat.nona, family =
binomial)
summary(res.log)

## Performance et diagnostique du modèle
check_model(res.log)
tbl_regression(res.log, exponentiate = TRUE)

## La fonction cut
crp <- c(10, 15, 14, 20, 45, 100, 150)
cut(x = crp,
    breaks = c(0, 10, 20, 50, 200),
    labels = c("<10", "(10,20]", "(20,50]", ">50"))

## Analyse des données nutrimentu

## Packages
library(ggcorrplot)
library(ggplot2)
library(readxl)

## Import des données
nutrimenu <- read_excel("nutrimenu.xlsx")

ggplot(
  data = nutrimentu,
  mapping = aes(
    x = Nutriscore,
    y = Energie,
    color = Type)) +
  geom_boxplot() +
  geom_point(
    position = position_dodge(width = .75)) +
  theme_minimal() +
  labs(title = "Energie par Nutriscore",

```

```

x = "Nutriscore",
y = "Energie (kcal)")

# Sélectionner les variables numériques
variables_nutritionnelles <- c("Glucides", "Lipides", "Proteines",
"Sucre", "Fibres", "Sel", "AGS", "Energie")

# Calculer la matrice de corrélation
correlation_matrix <- cor(nutrimenu[, variables_nutritionnelles])

# Afficher la matrice de corrélation
print(correlation_matrix)

# Créer le corrélogramme
ggcorrplot(correlation_matrix,
            lab = TRUE,      # Ajouter les valeurs des corrélations
dans le graphique
            lab_size = 3,    # Taille des étiquettes
            type = "lower",  # Afficher uniquement la partie
inférieure de la matrice (pour éviter la duplication)
            colors = c("red", "white", "blue"), # Palette de
couleurs
            title = "Corrélogramme des corrélations entre les
variables nutritionnelles",
            ggtheme = theme_minimal()) # Thème minimal

## ANOVA de l'énergie en fonction du Nutriscore
anova_Nutriscore <- aov(Energie ~ Nutriscore, data = nutrimenu)
summary(anova_Nutriscore)
pairwise.t.test(
  x = nutrimenu$Energie,
  g = nutrimenu$Nutriscore)

performance::check_model(regression_model_multiple)
gtsummary::tbl_regression(regression_model_multiple)

```



# Dates

```
library(lubridate)
dat1 <- ymd("2020-07-13")
dat2 <- ymd("2024-03-14")

interval_period <- interval(dat1, dat2)
full_year <- interval_period %/% years(1)
remaining_months <- interval_period %% years(1) %/% months(1)
remaining_days <- interval_period %% years(1) %% months(1) %/%
days(1)
sprintf('Your age is %d years, %d months and %d days', full_year,
remaining_months, remaining_days)
```