

MERCREDI MATIN

Installation du package du cours

```
remotes::install_github("vguillemot/intro2r")
remotes::install_github("vguillemot/intro2r", force
= TRUE)
```

Error: unexpected symbol in:

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

Error in loadNamespace(x) : aucun package nommé
'remotes' n'est trouvé

```
remotes::install_github("vguillemot/intro2r")
Downloading GitHub repo vguillemot/intro2r@HEAD
Error in utils::download.file(url, path, method =
method, quiet = quiet,  :
                                download from
'https://api.github.com/repos/vguillemot/intro2r/tar
ball/HEAD' failed
```

Exercice : booléens

- `1 == 2`
- `!(5 > -6)`
- `(1 <= 10) | (1 > 0)`

MERCREDI APRÈS-MIDI

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

Exercice

Extrayez la colonne de la teneur en sucres de la
table des fruits... de deux façons différentes !

Créez un objet contenant la teneur en sucres : quelle est la classe de cet objet ?

```
fruits$Sucres  
fruits[, "Sucres"]  
sucres <- fruits$Sucres
```

Exercice

Créez un vecteur **groupe** contenant les groupes de fruits. Donnez deux façons différentes d'extraire les dix premières valeurs de ce vecteur.

```
grp <- fruits$groupe  
grp[1:10]  
head(grp, 10)  
fruits[1:10, "groupe"]
```

Exercice

Construisez la sous-table contenant la teneur en protéines, en glucides et en lipides des fruits secs.

1. Sélection des colonnes de la teneur en protéines, en glucides et en lipides

```
fruits[, c("Glucides", "Proteines", "Lipides")]
```

2. Sélection des fruits secs

```
fruits[fruits$groupe == "secs", ]
```

3. On combine les deux

```
fruits_secs3col <- fruits[fruits$groupe == "secs",  
c("Glucides", "Lipides", "Proteines")]
```

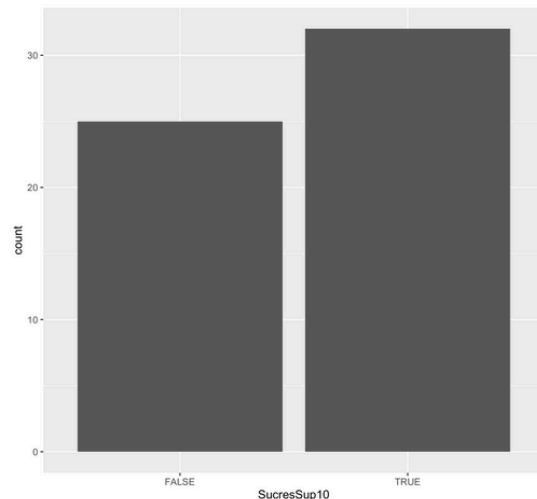
Diagramme en bâtons

```
barplot(table(fruits$groupe))
```

A vous !

À quoi servent les deux premières lignes ? Comment corriger la commande suivante pour obtenir le graphe à droite ?

```
fruits$SucreSup10 <-  
  fruits$Sucre > 10  
  
ggplot(***,  
  aes(***,  
    fill = SucreSup10)) +  
  geom_***()
```



```
fruits$SucreSup10 <- fruits$Sucre > 10
```

```
ggplot(data=fruits,  
  mapping = aes(x = SucreSup10 ,  
    fill = SucreSup10)) +  
  geom_bar()
```

Diagrammes en bâtons 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, aes(x = groupe, y =  
  VitamineC)) +  
  geom_col()
```

Créer le vecteur des moyennes en Vitamine C

```
tapply(fruits$VitamineC, fruits$groupe, FUN = mean)
```

Exemple avec des chats et des chiens

```
exemple <- data.frame(  
  race = c("chien", "chien", "chat", "chat"),  
  taille = c(10, 4, 16, 87))
```

exemple

```
ave(exemple$race, exemple$taille, FUN = mean) # 4  
valeurs  
tapply(exemple$taille, exemple$race, FUN = mean) # 2  
valeurs
```

```
moyennes <- tapply(fruits$VitamineC, fruits$groupe,  
FUN = mean)
```

```
vitamineC2 <- data.frame(  
  groupe = names(moyennes),  
  VitamineC = moyennes)
```

```
ggplot(  
  data = vitamineC2,  
  aes(x = groupe, y = VitamineC)) +  
  geom_col()
```

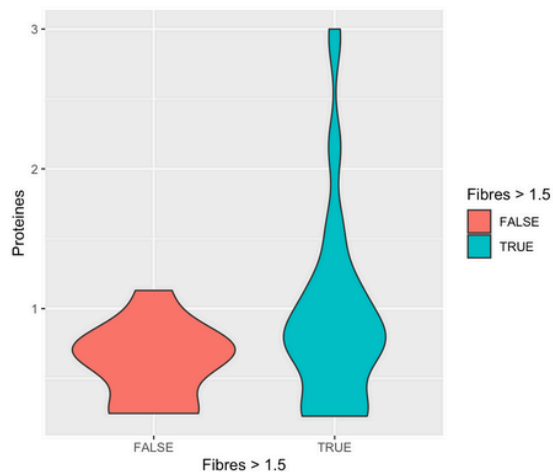
Histogramme

```
ggplot(fruits, aes(Sucres)) +  
  geom_histogram(breaks = seq(0, 75, 5))
```

A vous !

Complétez le code suivant pour obtenir la figure suivante :

```
ggplot(fruits,  
      aes(x = Fibres > 1.5,  
          y = Proteines,  
          fill = ***) ) +  
geom_***()
```



```
ggplot(fruits,  
      aes(x = Fibres > 1.5,  
          y = Proteines,  
          fill = Fibres > 1.5)) +  
geom_violin()
```

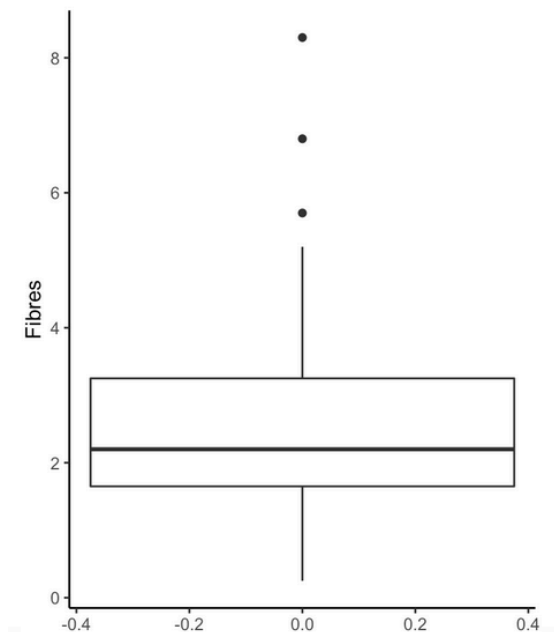
Personnalisation avec theme_bw()

```
ggplot(fruits, aes(Fibres)) +  
  geom_histogram() +  
  theme_bw()
```

A vous !

1. Consultez la page d'aide de la fonction `theme_bw` avec la commande `?theme_bw`
2. Choisissez la thème permettant de réaliser le graphe à droite en complétant la commande suivante.

```
ggplot(fruits, aes(y = Fibres)) +  
  geom_boxplot() +  
  theme_***()
```



```
ggplot(fruits, aes(y = Fibres)) +  
  geom_boxplot() +  
  theme_classic()
```

Personnalisation des titres

```
ggplot(fruits, aes(y = Fibres)) +  
  geom_boxplot() +  
  theme_classic() +  
  labs(  
    y = "Teneur en Fibres",  
    title = "Diagramme de Tukey")
```

JEUDI MATIN

```
library(ggplot2)
```

```
# Pour installer la librairie "intro2r"
```

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

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

Nuage de points :

```
ggplot(fruits, aes(x = Phosphore, y = Calcium, size =  
Magnesium)) +  
  geom_point()
```

ERREURS :

```
Error in loadNamespace(x) : there is no package  
called 'remotes'
```

SOLUTION -> installer "remotes"

```
Error in ggplot(fruits, aes(x = Phosphore, y =  
Calcium, size = Magnesium)) :  
  object 'fruits' not found
```

EXPLICATION -> j'ai oublié de charger les données
fruits

SOLUTION -> j'exécute la ligne de commande "data..."

```
# Et si c'était la couleur le Magnésium ?
```

```
ggplot(fruits,  
  aes(x = Phosphore,
```

```

        y = Calcium,
        color = Magnesium)) +
geom_point()

ggplot(fruits, aes(x = Phosphore,
                    y = Calcium,
                    color = groupe)) +
geom_point()

```

Error: object 'groupe' not found

Amel:

```

#couleur des points en fonction des groupes et
pondérée par teneur Mg
ggplot(fruits, aes(x = Phosphore, y = Calcium, size =
Magnesium, color = groupes)) +
  geom_point()

```

```

Error in `geom_point()`:
! Problem while computing aesthetics.
i Error occurred in the 1st layer.
Caused by error in `FUN()`:
! object 'groupes' not found
Run `rlang::last_trace()` to see where the error
occurred.

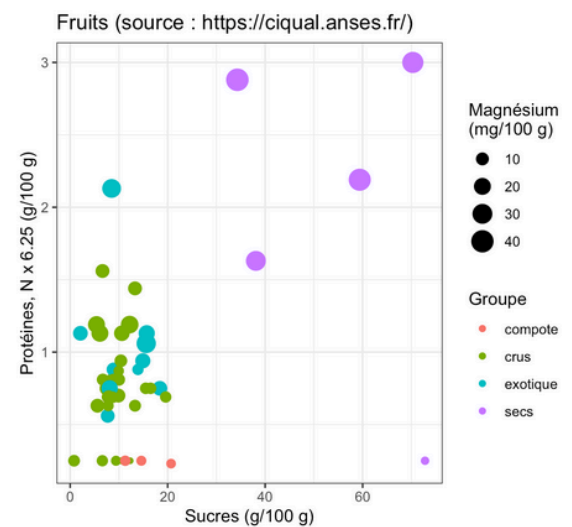
```

SOLUTION -> enlever le "s" de "groupes"

A vous !

Complétez la commande suivante pour obtenir le graphe ci-contre.

```
ggplot(fruits,
      aes(x = Sucres,
          y = Proteines,
          *** = Magnesium,
          *** = ***) +
  geom_***() +
  ***(title = "Fruits",
      x = "Sucres (g/100 g)",
      y = "Protéines, N x 6.25
(g/100 g)",
      size = "Magnésium\n(mg/100
g) ",
      ***= "Groupe") +
  theme_***()
```



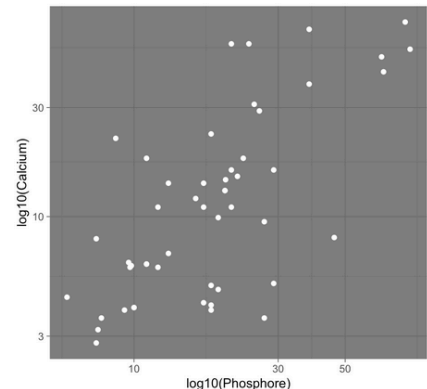
Proposition

```
ggplot(fruits,
      aes(x = Sucres,
          y = Proteines,
          size = Magnesium,
          color = groupe)) +
  geom_point() +
  labs (title = "Fruits",
        x = "Sucres (g/100 g)",
        y = "Protéines, N x 6.25 (g/100 g)",
        size = "Magnésium\n(mg/100 g)",
        color= "Groupe") +
  theme_bw()
```

A vous !

Complétez la commande suivante pour obtenir le graphe ci-contre.

```
ggplot(fruits,  
       aes(Phosphore,  
           Calcium)) +  
  geom_point(*** = "white") +  
  scale_***() +  
  scale_***() +  
  labs(x = "log10(Phosphore)",  
       y = "log10(Calcium)") +  
  theme_dark()
```



Error in log10() : 0 arguments passed to 'log10'
which requires 1

SOLUTION : utiliser `scale_x_log10` plutôt que
`scale_type` qui n'existait pas

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

Exemple de l'utilisation de "breaks" dans un histogramme et sur l'axe des abscisses

```
ggplot(fruits, aes(Sucres)) +  
  geom_histogram(breaks = seq(0, 80, 10)) +  
  scale_x_continuous(breaks = c(7, 26, 100))
```

```
ggplot(fruits,  
  aes(Phosphore,  
    Calcium,  
    color = "le point")) +  
  geom_point() +  
  labs(color = "Le titre") +  
  scale_color_manual(  
    values = c("le point" = "grey"))
```

JEUDI APRES-MIDI

```
# Import des données: trois façons  
nutri1 <- read.table(  
  "nutrimenu.txt",  
  header = TRUE)  
nutri2 <- read.csv("nutrimenu.csv", sep = ";")  
nutri3 <- readxl::read_excel("nutrimenu.xlsx")  
  
## Si on a oublié les commandes : utiliser  
l'interface de RSTUDIO  
  
# Import des données  
nutri <- read.table("nutrimenu.txt", header = TRUE)  
nutri_num <- nutri[, -(1:4)]
```

```

-(1:4) # Parenthèse mathématiques
-c(1, 2, 3, 4) # Parenthèses de fonction

## Chargement des librairies
library(pheatmap)
library(ggplot2)

## Nuage de points

ggplot(nutri,
       aes(x = Glucides,
           y = Lipides,
           color= Type,
           size = Energie)) +
  geom_point() +
  theme_classic() +
  labs(
    title = "Composition énergétique des plats",
    x = "Teneur en Glucides",
    y = "Teneur en Lipides",
    color = "",
    size = "Energie (kCal)"
  )

## Diagrammes de Tukey

ggplot(
  data = nutri,
  mapping = aes(x = Type, y = Energie)) +
  geom_boxplot() +
  geom_point(mapping = aes(color = Nutriscore)) +
  theme_bw()

## Histogrammes

```

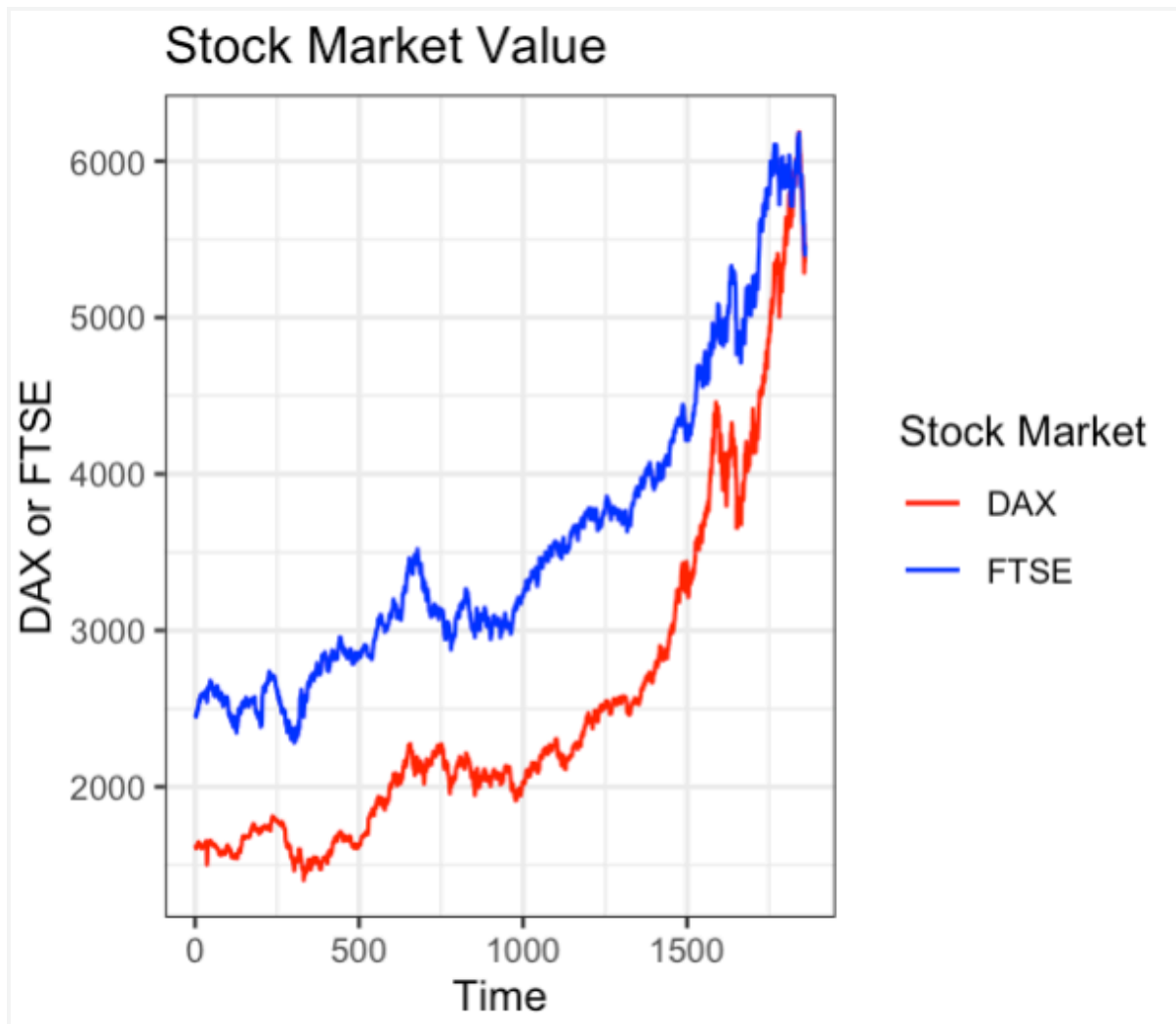
```
ggplot(nutri, aes(Glucides, fill = Type)) +  
  geom_histogram(bins = 10) +  
  facet_grid(Type~.) +  
  theme_bw() +  
  theme(legend.position = "bottom") +  
  labs(  
    x = "Teneur en Glucides",  
    y = "Nombre de Recettes",  
    fill = "")
```

Autre exemple :

```
dat <- data.frame(  
  x = 1:nrow(EuStockMarkets),  
  EuStockMarkets)
```



```
ggplot(dat, aes(x = x)) +  
  geom_line(aes(y = DAX, color = "DAX")) +  
  geom_line(aes(y = FTSE, color = "FTSE")) +  
  scale_color_manual(  
    values = c("DAX" = "red", "FTSE" = "blue")) +  
  theme_bw() +  
  labs(  
    title = "Stock Market Value",  
    x = "Time",  
    y = "DAX or FTSE",  
    color = "Stock Market")
```



```
## Autre exemple : epicurve
```

```
cli <- data.frame(  
  temps = 1:20,  
  conf = rbinom(n = 20, size = 10, prob = 0.4) ,  
  susp = rbinom(n = 20, size = 20, prob = 0.1) ,  
  prob = rbinom(n = 20, size = 30, prob = 0.1)  
)
```

```
library(tidyr)
```

```
cli_long <- pivot_longer(cli, -1, names_to = "type")
```

```

ggplot(cli, aes(x = temps)) +

  geom_col(aes(y = conf), fill = "orange")

ggplot(cli_long,
  aes(x = temps,
    y = value,
    fill = type)) +
  geom_col(aes(group = temps), color = "white")

ggplot(cli_long, aes(x = temps, y = value, color =
type)) +
  geom_point() +
  geom_line() +
  theme_bw()

## Pour choisir un test :
https://biostatgv.sentiweb.fr

#### Comparaison de l'Energie des fruits

library(ggplot2)

data("fruits", package = "intro2r")

# Graphe
ggplot(fruits, aes(groupe, Energie, color = groupe))
+
  geom_boxplot() +
  geom_point() +
  theme_bw()

```

```
## Un test d'homogénéité des variances
bartlett.test(Energie ~ groupe, data = fruits) #
paramétrique
fligner.test(Energie ~ groupe, data = fruits) # non
paramétrique
## Une ANOVA non paramétrique
kruskal.test(Energie ~ groupe, data = fruits)
## La procédure post-hoc
pairwise.wilcox.test(fruits$Energie, fruits$groupe)
```

Conclusion : les fruits secs sont significativement différents des fruits crus et exotique, sur la base d'un test de Kruskal Wallis ($p\text{-value} < 0.05$) suivi d'une procédure post-hoc basée sur le test de Wilcoxon, au risque alpha de 5%.

ILRpasteur2023