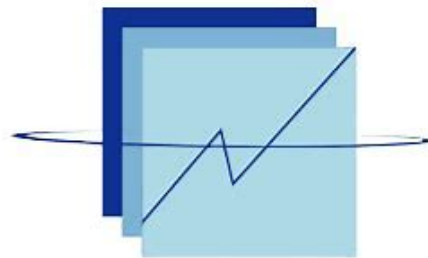


RÉPUBLIQUE DU SÉNÉGAL

Un Peuple - Un But - Une Foi

MINISTÈRE DE L'ÉCONOMIE
DU PLAN ET DE LA COOPÉRATION



ANSD

Agence nationale de la Statistique et de la Démographie (ANSD)



Ecole nationale de la Statistique et de l'Analyse économique Pierre Ndiaye (ENSAE)

Semestre 2 : Projet statistique sous R

Partie 1 : Introduction à Rmarkdown

Rédigé par :

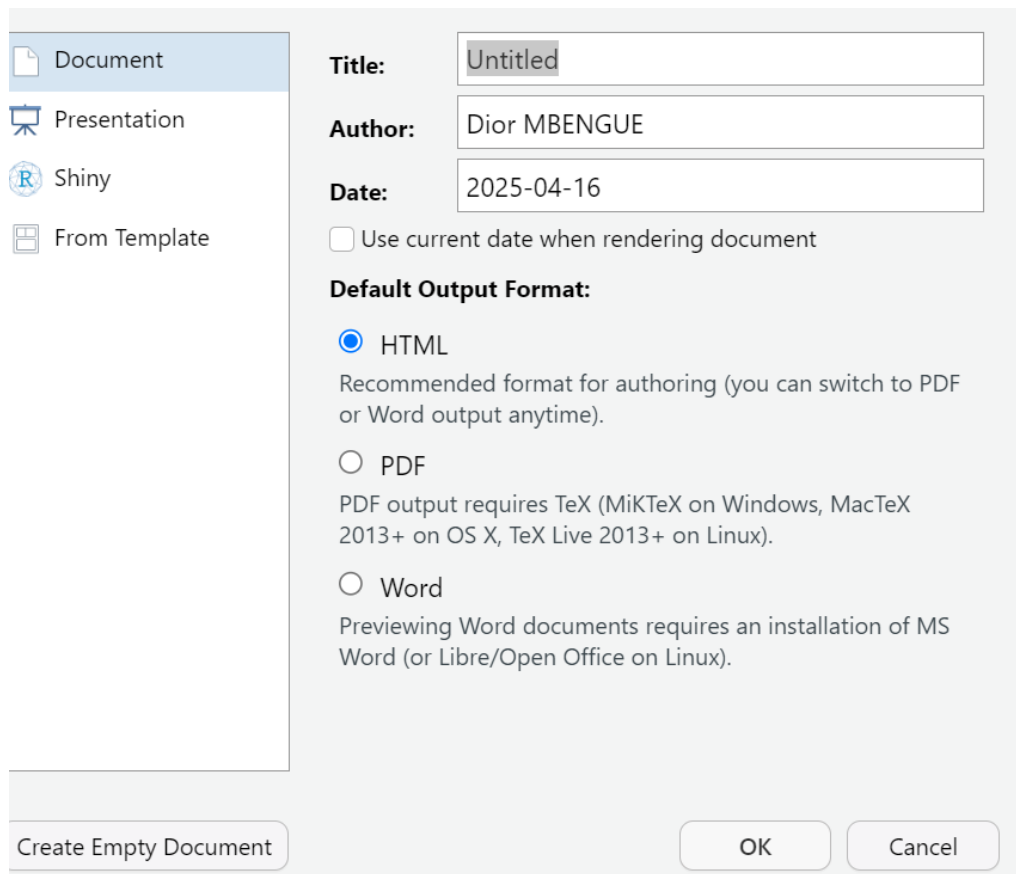
Khadidiatou Diakhaté
Haba Fromo Francis
Dior Mbengue

Sous la supervision de :

M. Aboubacre HEMA
Research Analyst

Année scolaire : 2024/2025

Créer un fichier Rmarkdown



The image shows the 'Create Empty Document' dialog box in RStudio. On the left, there is a sidebar with four options: 'Document' (selected), 'Presentation', 'Shiny', and 'From Template'. The main area contains fields for 'Title' (Untitled), 'Author' (Dior MBENGUE), and 'Date' (2025-04-16). Below these fields is a checkbox labeled 'Use current date when rendering document' which is unchecked. Under the heading 'Default Output Format:', there are three radio buttons: 'HTML' (selected), 'PDF', and 'Word'. Each radio button has a descriptive text block below it. The 'HTML' block says 'Recommended format for authoring (you can switch to PDF or Word output anytime)'. The 'PDF' block says 'PDF output requires TeX (MiKTeX on Windows, MacTeX 2013+ on OS X, TeX Live 2013+ on Linux)'. The 'Word' block says 'Previewing Word documents requires an installation of MS Word (or Libre/Open Office on Linux)'. At the bottom, there are three buttons: 'Create Empty Document', 'OK', and 'Cancel'.

Title: Untitled

Author: Dior MBENGUE

Date: 2025-04-16

☐ Use current date when rendering document

Default Output Format:

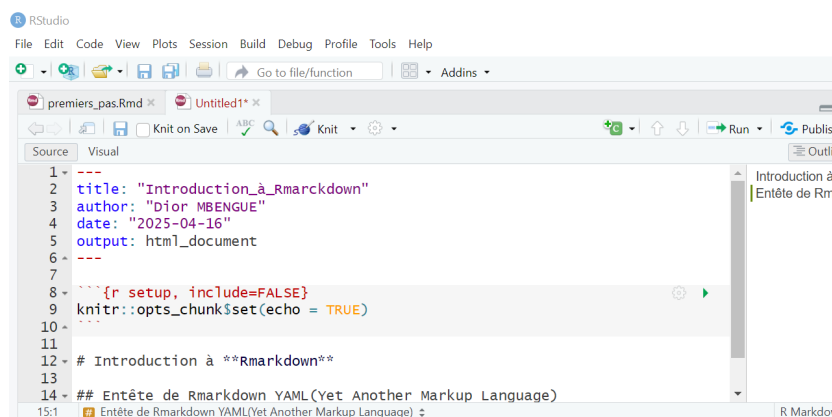
☒ HTML
Recommended format for authoring (you can switch to PDF or Word output anytime).

☐ PDF
PDF output requires TeX (MiKTeX on Windows, MacTeX 2013+ on OS X, TeX Live 2013+ on Linux).

☐ Word
Previewing Word documents requires an installation of MS Word (or Libre/Open Office on Linux).

Create Empty Document OK Cancel

Entête de Rmarkdown YAML(Yet Another Markup Language)

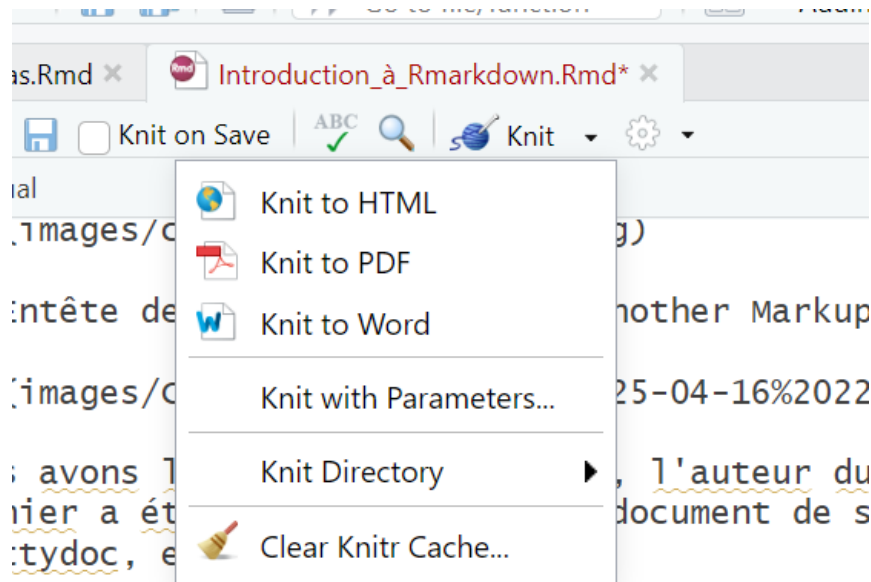


The image shows the RStudio editor interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. Below the menu bar is a toolbar with icons for saving, running, and other functions. The main editor window shows a file named 'premiers_pas.Rmd' with a YAML header. The header is enclosed in a dashed box and contains the following text: 'title: "Introduction_à_Rmarkdown"', 'author: "Dior MBENGUE"', 'date: "2025-04-16"', and 'output: html_document'. Below the header, there is a code chunk with the following text: 'r_setup, include=FALSE', 'knitr::opts_chunk\$set(echo = TRUE)'. The editor also shows a preview of the document, which displays the title 'Introduction à Entête de Rm'.

```
1 ---
2 title: "Introduction_à_Rmarkdown"
3 author: "Dior MBENGUE"
4 date: "2025-04-16"
5 output: html_document
6 ---
7
8 {r_setup, include=FALSE}
9 knitr::opts_chunk$set(echo = TRUE)
10
11
12 # Introduction à **Rmarkdown**
13
14 ## Entête de Rmarkdown YAML(Yet Another Markup Language)
15
```

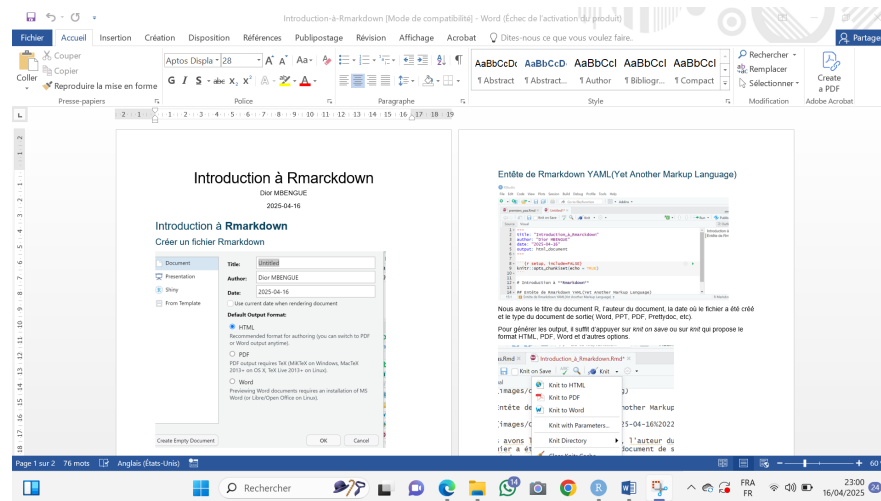
Nous avons le titre du document R, l'auteur du document, la date où le fichier a été créé et le type du document de sortie (Word, PPT, PDF, Prettydoc, etc).

Pour générer les output, il suffit d'appuyer sur *knit on save* ou sur *knit* qui propose le format HTML, PDF, Word et d'autres options.

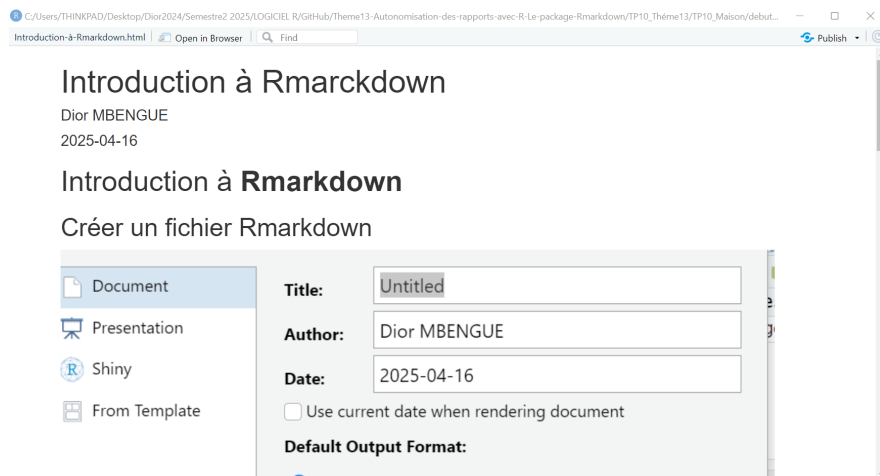


Pour générer les output, il suffit d'appuyer sur *knit on save* ou sur *knit* qui propose le format HTML, PDF, Word et d'autres options.

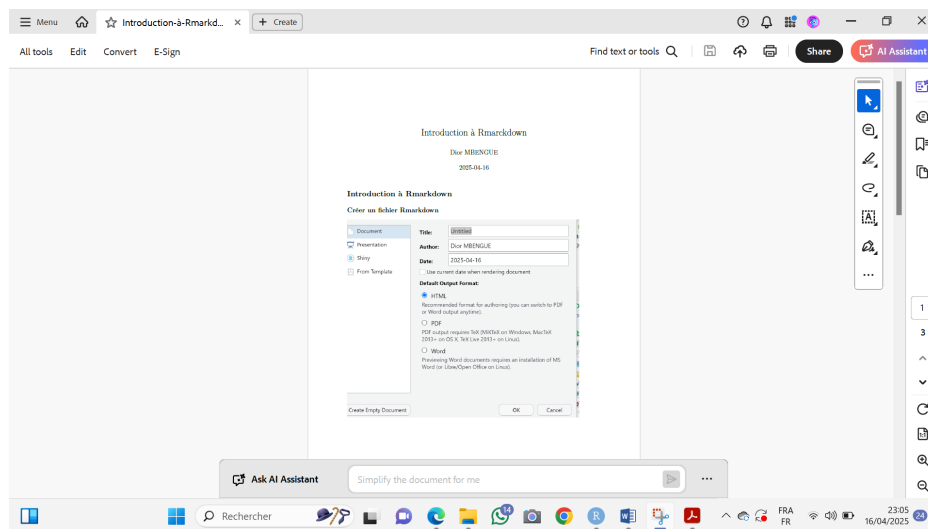
OUTPUT Word



OUTPUT HTML



OUTPUT PDF



Pour avoir la sortie PDF, il faut installer LaTeX ou Mictex

Nous avons aussi l'output **Prettydoc**

Les chunk options et les global options

chunk de configuration

Il est préférable de configurer le comportement par défaut Knitr pour faire certains réglages comme la position et la taille des figures ou l'affichage des lignes de codes (instructions sur R) ou non dans le document de sortie. Ce chunk de configuration utilise cette fonction pour définir les options par défaut applicables à tous les chunk en dessous.

```
knitr::opts_chunk$set()
```

Après ce chunk, il est recommandé de faire un autre chunk pour charger les packages et un autre pour charger les données qui seront utilisés tout au long.

```
#vérifier et installer les packages nécessaires
packages <- c("haven", "utils", "dplyr", "gtsummary", "survey", "labelled", "compareDF")
for (pkg in packages) {
  if (!require(pkg, character.only = TRUE)) install.packages(pkg, dependencies = TRUE)
  library(pkg, character.only = TRUE)
}
```

Dans la mesure du possible, il est préférable d'utiliser des chemins relatifs pour les fichiers (exemple : ./donnees/mesdonnees.csv), plutôt que des chemins absolus (exemple : D:/chemin/vers/les/donnees/mesdonnees.csv).

Il peut arriver qu'il soit difficile d'accéder à des fichiers en utilisant uniquement des chemins relatifs. Une solution de repli consiste à définir le chemin absolu du dossier à un seul endroit dans le code source du rapport, puis à l'utiliser de façon relative dans les fonctions d'importation. Voici un exemple :

```
# Définir UNE SEULE FOIS le répertoire des données
dossier_donnees <- "D:/chemin/vers/les/donnees/"
# Charger les données avec un chemin composé
donnees1 <- fread(paste0(dossier_donnees, "mesdonnees1.csv"))
donnees2 <- fread(paste0(dossier_donnees, "mesdonnees2.csv"))
```

On peut aussi nommer un chunk

```
resultat <- 1 + 1
print(resultat)
```

```
## [1] 4
```

```
## [1] 4
```

```
2+2
# eval=FALSE veut dire qu'on montre les lignes de code mais pas l'output
```

```
2+2
# eval=FALSE veut dire qu'on montre les lignes de code mais pas l'output
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 -
-
## v dplyr      1.1.4      v readr      2.1.5
## v forcats    1.0.0      v stringr    1.5.1
## v ggplot2     3.5.1      v tibble     3.2.1
## v lubridate  1.9.4      v tidyr      1.3.1
## v purrr       1.0.2
## -- Conflicts ----- tidyverse_conflicts() -
-
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to
```

```
# Pour que les messages générés sur le chunk ne soient pas affichés
```

```
library(tidyverse)
```

```
# Pour que les messages générés sur le chunk ne soient pas affichés
```

```
library(tidyverse)
```

```
## Warning: le package 'tidyverse' a été compilé avec la version R 4.4.3
```

```
## — Attaching core tidyverse packages — tidyverse 2.0.0 —
## ✓ forcats 1.0.0 ✓ readr 2.1.5
## ✓ ggplot2 3.5.1 ✓ stringr 1.5.1
## ✓ lubridate 1.9.4 ✓ tibble 3.2.1
## ✓ purrr 1.0.4 ✓ tidyr 1.3.1
## — Conflicts — tidyverse_conflicts() —
## ✖ tidy::expand() masks Matrix::expand()
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag() masks stats::lag()
## ✖ tidy::pack() masks Matrix::pack()
## ✖ tidy::unpack() masks Matrix::unpack()
## ! Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
# Pour que les messages générés sur le chunk ne soient pas affichés
```

```
library(tidyverse)
```

```
# Pour que les messages générés sur le chunk ne soient pas affichés
```

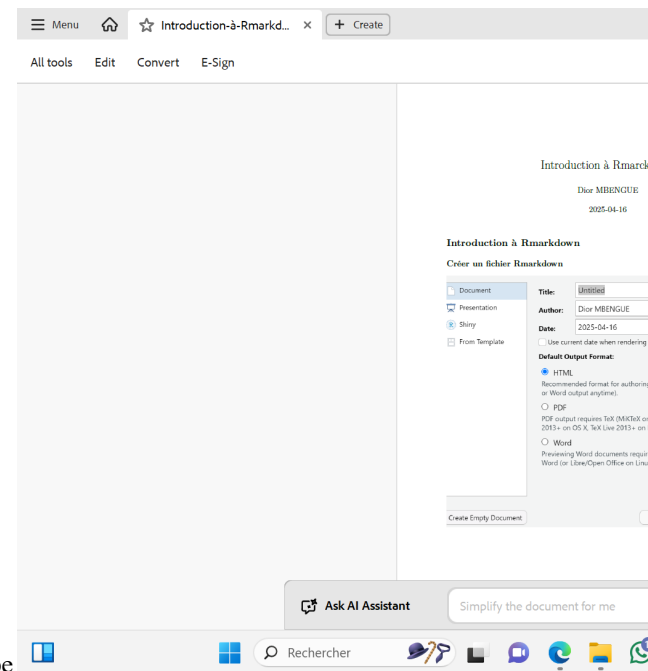
```
```{r configuration, include=TRUE}
knitr::opts_chunk$set(echo = FALSE,
 warning = FALSE,
 message = FALSE,
 error = TRUE,
 fig.align = "center",
 out.width = "75%")
```
```

Dans cet exemple, il est demandé à R Markdown de ne pas inclure les instructions R dans le document de sortie (echo = FALSE). Les warnings et les messages d’informations n’apparaîtront pas non plus dans le fichier de sortie. En revanche, les erreurs apparaîtront (error = TRUE). Les figures sont centrées (fig.align = “center”) et ont une largeur de 75% de la largeur du texte (out.width = “75%”).

Nous avons encore

```
```{r setup, include=FALSE}
knitr::opts_chunk$set(
 warning = FALSE, # Ne montre pas les avertissements
 results = "markup", # Type de rendu des résultats ('hide', 'asis', 'hold', 'markup')
 fig.align = "center", # Aligne les graphiques au centre
 fig.width = 6, # Largeur des graphiques
 fig.height = 4, # Hauteur des graphiques
 cache = FALSE, # Active ou non la mise en cache
 comment = NA # Supprime le préfixe '##' dans la sortie_des_résultats
)
```
```

Insertion d'éléments et de bullet



- Pour insérer une image sur Rmarkdown, on peut mettre un code du type
ou juste copier l'image dans la partie **Visual**
- Pour insérer un lien: `[(https://thegraph.org)]`

Inline Code

Les inlines codes dans Rmarkdown permettent d'insérer directement des résultats à l'intérieur du texte. Il sont utiles pour rendre le rapport dynamique.

```
max(women$height)
```

```
## [1] 72
```

The maximum women's height is 72.

```
## [1] 72
```

The maximum women's height is 72.

Les tableaux

```
women # sans recommandation de tableaux
```

```
##      height weight
## 1         58     115
## 2         59     117
## 3         60     120
## 4         61     123
## 5         62     126
## 6         63     129
## 7         64     132
## 8         65     135
## 9         66     139
## 10        67     142
## 11        68     146
## 12        69     150
## 13        70     154
## 14        71     159
## 15        72     164
```

```
women # sans recommandation de tableaux
```

```
##      height weight
## 1         58     115
## 2         59     117
## 3         60     120
## 4         61     123
## 5         62     126
## 6         63     129
## 7         64     132
## 8         65     135
## 9         66     139
## 10        67     142
## 11        68     146
## 12        69     150
## 13        70     154
## 14        71     159
## 15        72     164
```

```
# Afficher des tableaux à partir des données de R
pacman::p_load(flextable, gt, reactable)
flextable::flextable(women)
```


| height | weight |
|--------|--------|
| 58 | 115 |
| 59 | 117 |
| 60 | 120 |
| 61 | 123 |
| 62 | 126 |
| 63 | 129 |
| 64 | 132 |
| 65 | 135 |
| 66 | 139 |
| 67 | 142 |
| 68 | 146 |
| 69 | 150 |
| 70 | 154 |
| 71 | 159 |
| 72 | 164 |

```
# gt::gt(women)
# reactable::reactable(women)
```

Un autre exemple

```
head(iris)
```

```
##      Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1           5.1         3.5         1.4         0.2   setosa
## 2           4.9         3.0         1.4         0.2   setosa
## 3           4.7         3.2         1.3         0.2   setosa
## 4           4.6         3.1         1.5         0.2   setosa
## 5           5.0         3.6         1.4         0.2   setosa
## 6           5.4         3.9         1.7         0.4   setosa
```

```
# cela affichera les premières lignes de iris
```

```
knitr::kable(head(iris), caption = "Premières lignes du jeu de données_iris")
```

Table 2: Premières lignes du jeu de données_iris

| Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
|--------------|-------------|--------------|-------------|---------|
| 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| 4.9 | 3.0 | 1.4 | 0.2 | setosa |
| 4.7 | 3.2 | 1.3 | 0.2 | setosa |

| Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
|--------------|-------------|--------------|-------------|---------|
| 4.6 | 3.1 | 1.5 | 0.2 | setosa |
| 5.0 | 3.6 | 1.4 | 0.2 | setosa |
| 5.4 | 3.9 | 1.7 | 0.4 | setosa |

```
# On peut améliorer les tableaux avec cette variable
```

```
library(kableExtra)
```

```
##
## Attachement du package : 'kableExtra'

## Les objets suivants sont masqués depuis 'package:flextable':
##
##     as_image, footnote

## L'objet suivant est masqué depuis 'package:dplyr':
##
##     group_rows

knitr::kable(head(iris), caption = "Iris avec kableExtra")
```

Table 3: Iris avec kableExtra

| Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
|--------------|-------------|--------------|-------------|---------|
| 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| 4.9 | 3.0 | 1.4 | 0.2 | setosa |
| 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| 4.6 | 3.1 | 1.5 | 0.2 | setosa |
| 5.0 | 3.6 | 1.4 | 0.2 | setosa |
| 5.4 | 3.9 | 1.7 | 0.4 | setosa |

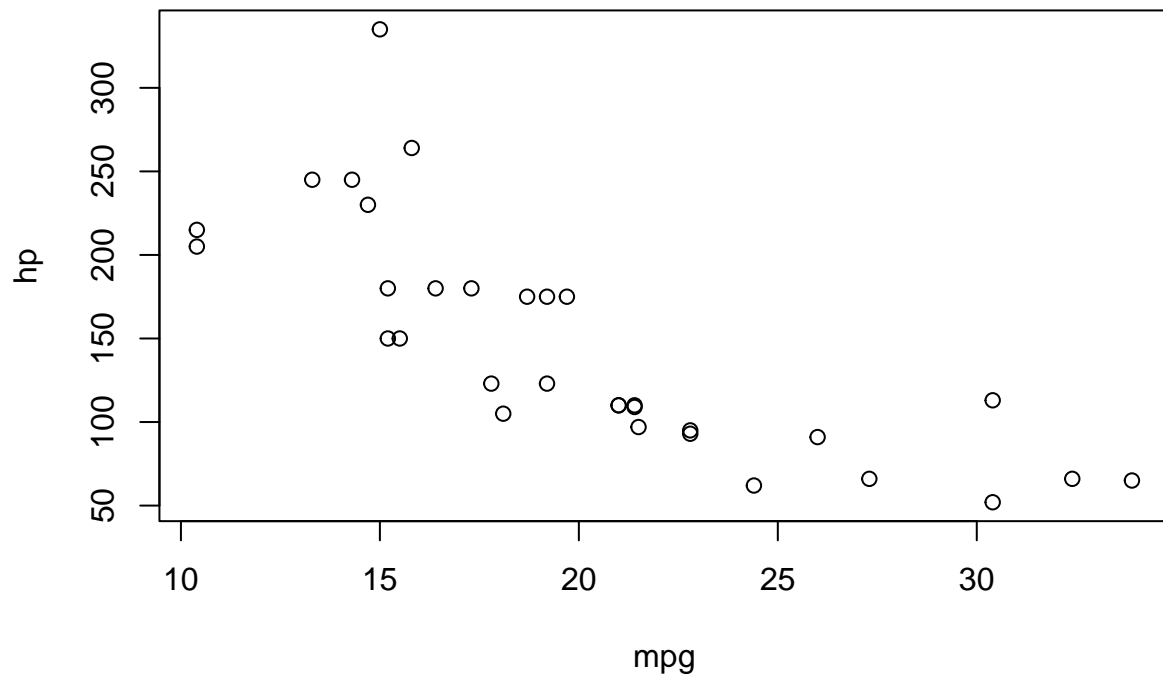
```
library(DT)
datatable(iris)
# Cela permet de filtrer, trier, rechercher dans le tableau (pour HTML uniquement)
```

Les graphiques

Il faut d'abord installer les packages ggplot2, cowplot, patchwork, plotly, etc.

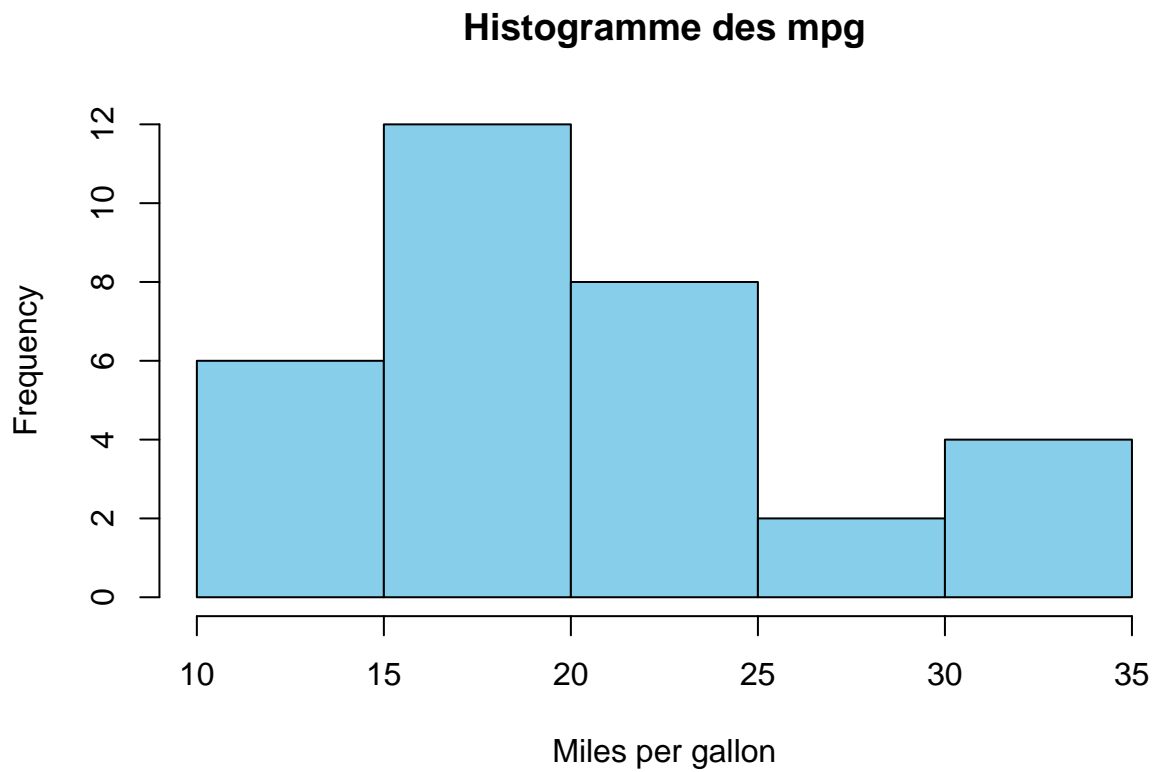
```
plot(mtcars$mpg, mtcars$hp,
     main = "Puissance vs Consommation",
     xlab = "mpg", ylab="hp")
```

Puissance vs Consommation



Cela produit un nuage de points

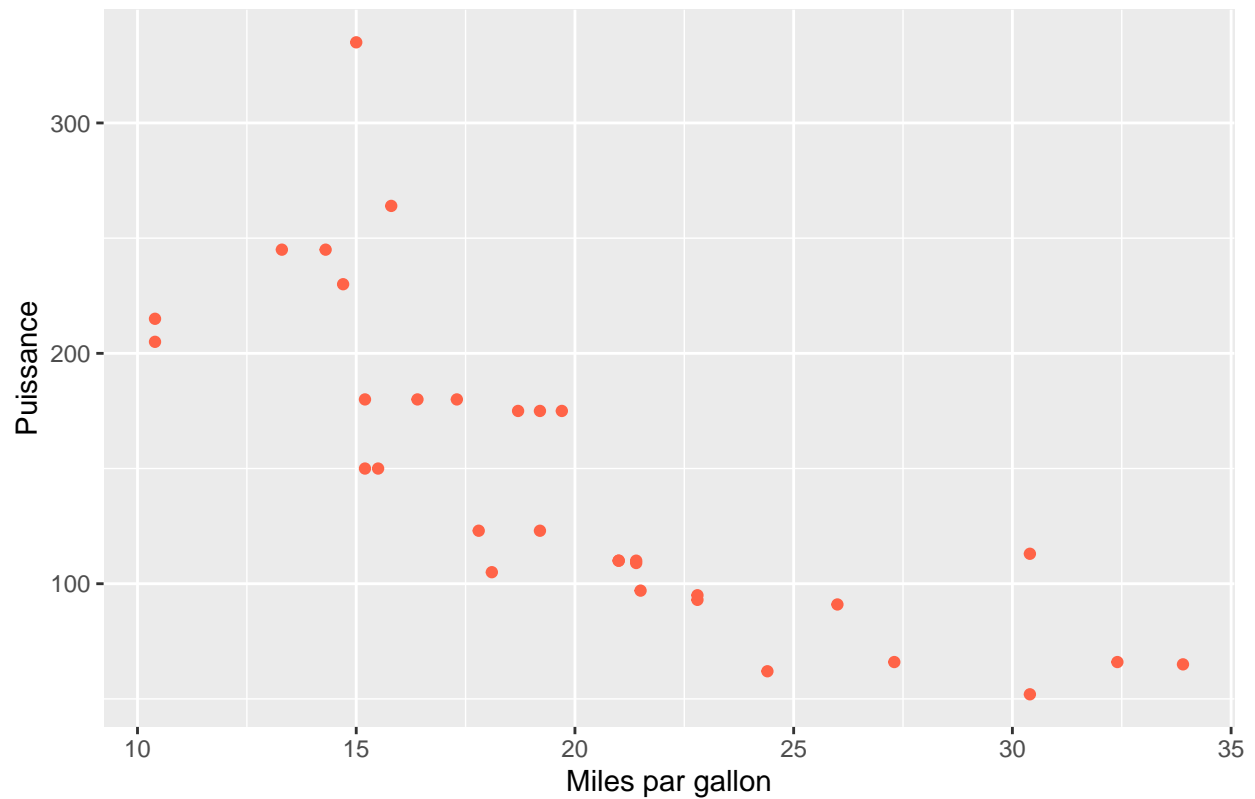
```
hist(mtcars$mpg, col = "skyblue",  
      main = "Histogramme des mpg",  
      xlab = "Miles per gallon")
```



Ceci est un histogramme

```
ggplot2::ggplot(mtcars, aes(x = mpg, y = hp)) +  
  geom_point(color = "tomato") +  
  labs(title = "Puissance en fonction du mpg",  
        x = "Miles par gallon", y="Puissance")
```

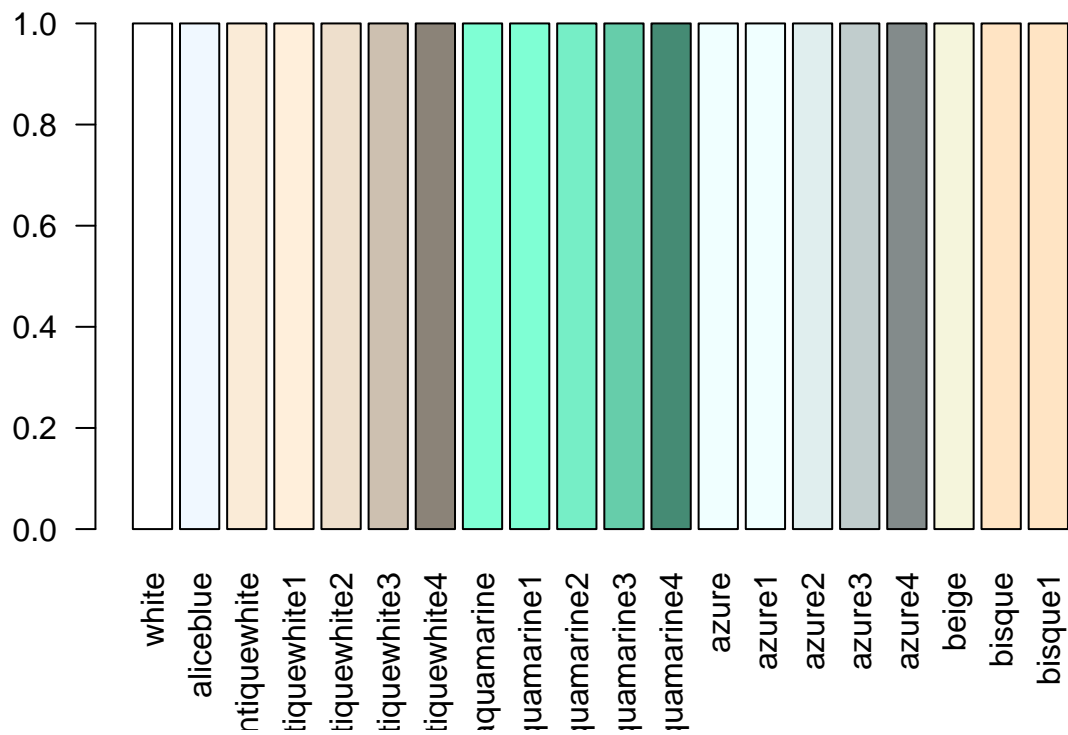
Puissance en fonction du mpg



```
#Avec le package ggplot2
```

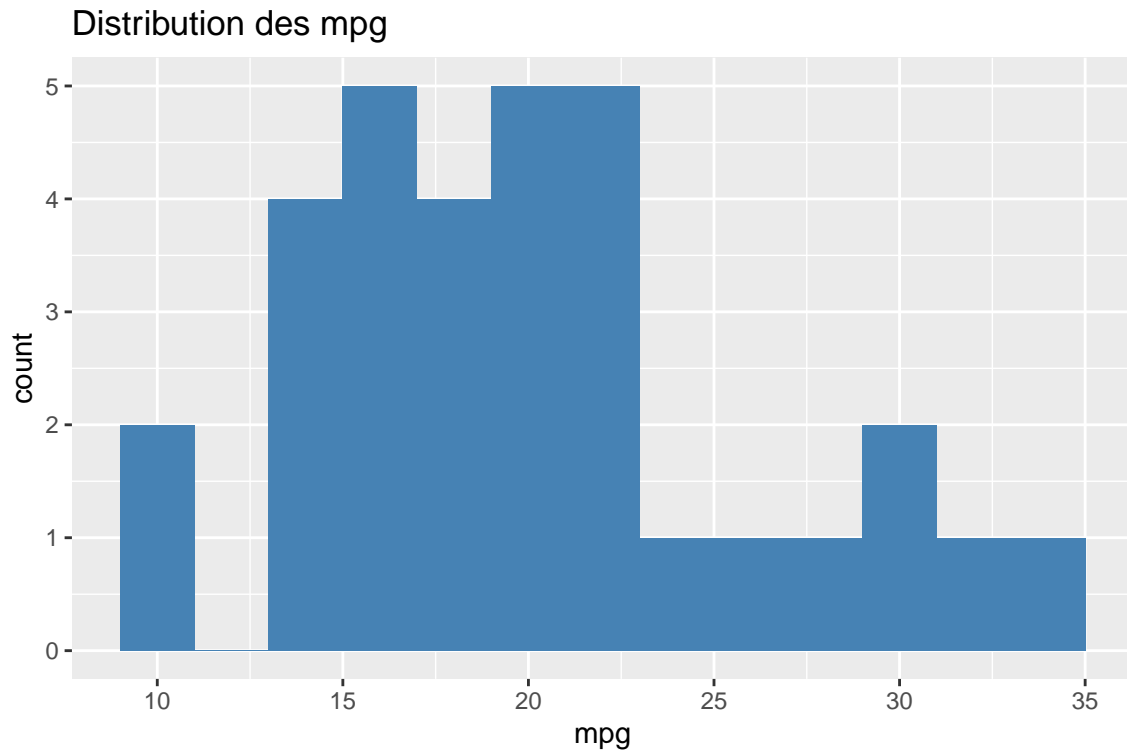
On peut mettre la couleur qu'on veut

```
# Affiche les 20 premières couleurs avec leur nom
barplot(rep(1, 20), col = colors()[1:20], names.arg = colors()[1:20], las=2)
```

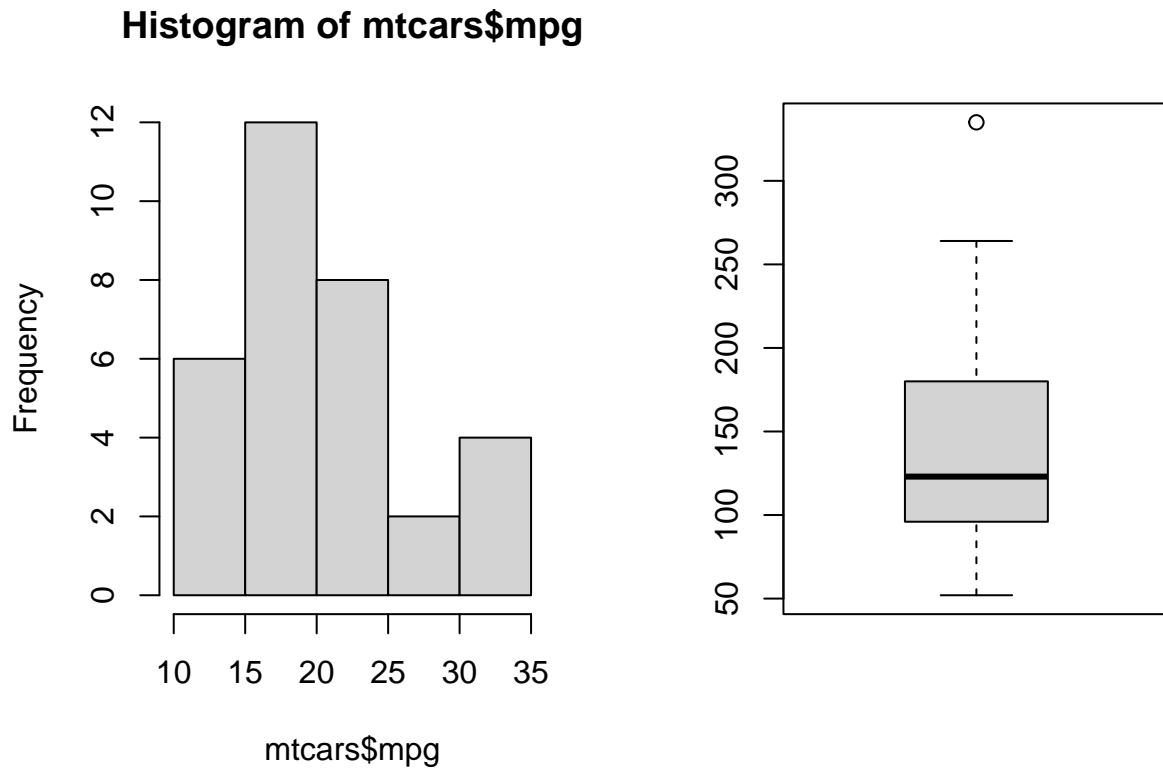


on peut aussi controler la taille et l'alignement

```
ggplot(mtcars, aes(x = mpg)) +  
  geom_histogram(binwidth = 2, fill = "steelblue") +  
  labs(title = "Distribution des mpg")
```



```
# Nous pouvons aussi avoir plusieurs graphiques cote à cote  
par(mfrow = c(1, 2)) # 1 ligne, 2 colonnes  
hist(mtcars$mpg)  
boxplot(mtcars$hp)
```



Pour avoir plus d'informations, vous pouvez partir dans [<https://r-graph-gallery.com/>]

Citation et bibliographie

Introduction

Selon @smith2020, la statistique est essentielle dans l'analyse des données.

Voir aussi l'étude de @doe2019 pour un autre point de vue.

Conclusion

La bibliographie s'affiche ci-dessous automatiquement.

Références