



# INTRODUCTION AU PACKAGE TIDYVERSE

Amandine Blin, UAR 2700, Pôle Analyse de Données

15/11/2023

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Le package tidyverse . . . . .	1
1.2	Le workflow . . . . .	2
1.3	Les ressources utiles . . . . .	2
1.4	Citer R et le package tidyverse . . . . .	2
1.5	Le jeu de données airquality . . . . .	3
<b>2</b>	<b>L'importation de données : les packages readr et readxl</b>	<b>5</b>
2.1	Présentation . . . . .	5
2.2	Exemples . . . . .	5
2.3	Exemple 1 : importation d'un fichier Excel . . . . .	5
2.4	Exemple 2 : importation d'un fichier .csv . . . . .	6
<b>3</b>	<b>Mise en forme et manipulation des données</b>	<b>7</b>
3.1	Qu'est ce qu'un tibble ? . . . . .	7
3.2	Quelques opérations élémentaires avec le package dplyr . . . . .	7
3.3	Jointure de deux tableaux . . . . .	17
3.4	Transformation de données : Le package tidyr . . . . .	18
<b>4</b>	<b>Les graphiques avec le package ggplot2</b>	<b>21</b>
4.1	Le principe . . . . .	21
4.2	Exemple d'un nuage de points . . . . .	21
4.3	Ajouter des éléments statistiques . . . . .	37
4.4	Construction d'un histogramme et d'une courbe de densité . . . . .	40
4.5	Construction d'une boîte à moustaches . . . . .	42
4.6	Construction d'un diagramme en barres . . . . .	43
4.7	Les addins ggplotAssist et esquisse . . . . .	44
<b>5</b>	<b>Un petit détour par le package broom</b>	<b>46</b>
5.1	Pourquoi utiliser le package broom ? . . . . .	46
5.2	Résumer la sortie d'un modèle statistique : La fonction tidy() . . . . .	46
5.3	Récapitulatif d'un modèle : La fonction glance() . . . . .	46
5.4	Ajouter des informations sur le tableau de données : La fonction augment() . . . . .	47
5.5	Pour aller plus loin . . . . .	47

# Chapitre 1

## Introduction

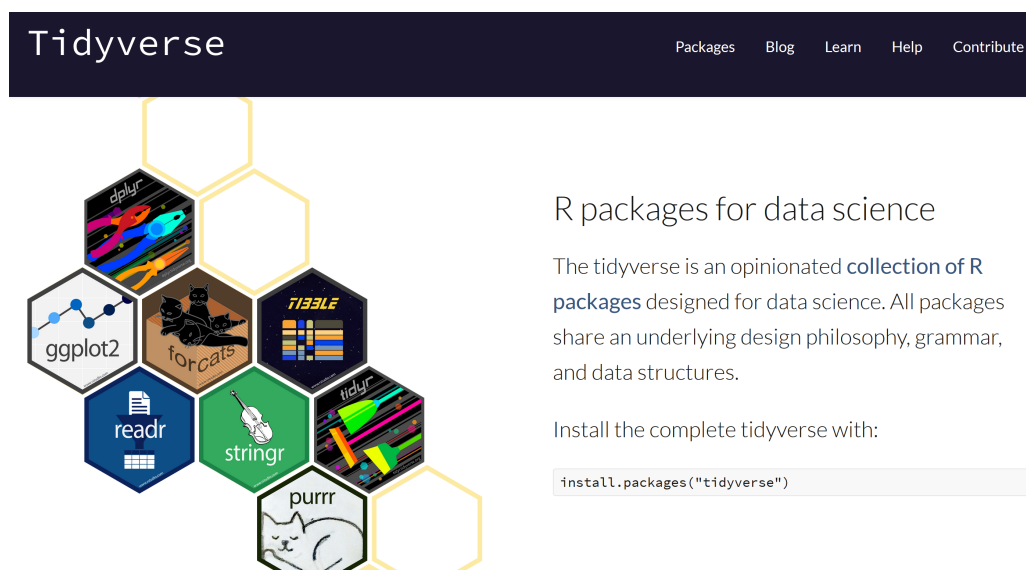
### 1.1 Le package tidyverse

- Regroupement de différents packages s'appuyant sur la même grammaire et structuration de données (WICKHAM et al. [8])
- Il existe une importante communauté d'utilisateurs et d'aide (<https://www.tidyverse.org/>).

```
library(tidyverse)
```

Lorsqu'on appelle la librairie tidyverse, d'autres librairies sont chargées.

- Le package readr : Lire des données tabulaires (.csv)
- Le package tibble : Travailler sur la version moderne des tableaux de données
- Le package dplyr : Ensemble d'outils pour manipuler des données
- Le package tidyr : Restructuration des données
- Le package stringr : Manipulation de chaînes de caractères
- Le package forcats : Manipulation de variables qualitatives (factor)
- Le package purrr : Ensemble d'outils pour travailler avec des fonctions et des vecteurs (programmation)
- Le package ggplot2 : Réalisation de graphiques



L'ensemble de ces packages constitue la base du data scientist.

## 1.2 Le workflow

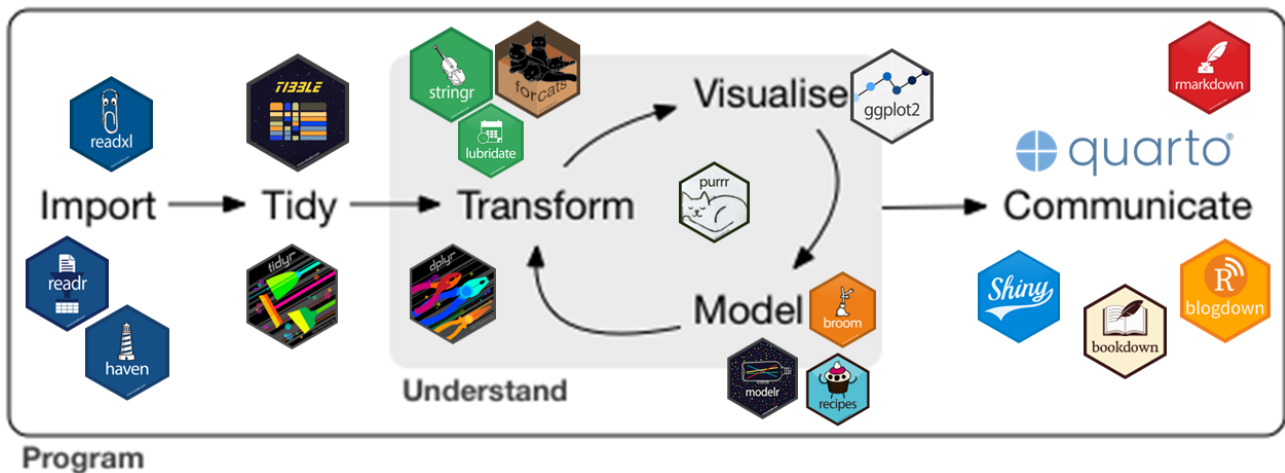


Figure 1.1: D'après "R for Data Science", H. Wickham & G. Grolemund, O'Reilly Media (2017)

## 1.3 Les ressources utiles

- *Introduction à R et au tidyverse* (<https://juba.github.io/tidyverse>), J. Barnier
- Manuels sur R disponibles sur le CRAN (<https://cran.r-project.org>)
- Documentation disponible sur RStudio (<https://rstudio.com/resources/cheatsheets/>)
- Livres : CHANG [2], WICKHAM [6], WICKHAM et GROLEMUND [5], WICKHAM [4]

## 1.4 Citer R et le package tidyverse

```
# Citation de R
citation()
```

To cite R in publications use:

```
R Core Team (2023). _R: A Language and Environment for Statistical
Computing_. R Foundation for Statistical Computing, Vienna, Austria.
<https://www.R-project.org/>.
```

Une entrée BibTeX pour les utilisateurs LaTeX est

```
@Manual{,
  title = {R: A Language and Environment for Statistical Computing},
  author = {{R Core Team}},
  organization = {R Foundation for Statistical Computing},
  address = {Vienna, Austria},
  year = {2023},
  url = {https://www.R-project.org/},
}
```

We have invested a lot of time and effort in creating R, please cite it when using it for data analysis. See also 'citation("pkgname")' for

citing R packages.

```
# Citation du package tidyverse
citation(package="tidyverse")
```

Pour citer le package 'tidyverse' dans une publication, utilisez :

Wickham H, Averick M, Bryan J, Chang W, McGowan LD, François R, Golemund G, Hayes A, Henry L, Hester J, Kuhn M, Pedersen TL, Miller E, Bache SM, Müller K, Ooms J, Robinson D, Seidel DP, Spinu V, Takahashi K, Vaughan D, Wilke C, Woo K, Yutani H (2019). "Welcome to the tidyverse." *\_Journal of Open Source Software\_*, \*4\*(43), 1686. doi:10.21105/joss.01686 <<https://doi.org/10.21105/joss.01686>>.

Une entrée BibTeX pour les utilisateurs LaTeX est

```
@Article{,
  title = {Welcome to the {tidyverse}},
  author = {Hadley Wickham and Mara Averick and Jennifer Bryan and Winston Chang and I},
  year = {2019},
  journal = {Journal of Open Source Software},
  volume = {4},
  number = {43},
  pages = {1686},
  doi = {10.21105/joss.01686},
}
```

## 1.5 Le jeu de données airquality

Différentes mesures de la qualité de l'air à New York ont été relevées de mai à septembre 1973 (New York State Department of Conservation, National Weather Service, CHAMBERS et al. [1]) à l'aéroport de La Guardia. Le jeu de données a 153 observations et 6 variables :

- La variable *Ozone* : Taux d'ozone (parts per billion)
- La variable *Solar.R* : Rayonnement solaire
- La variable *Wind* : Vitesse du vent (miles per hour)
- La variable *Temp* : Température (degré Fahrenheit)
- La variable *Month* : Le numéro du mois (1 à 12)
- La variable *Day* : Le numéro du jour du mois (de 1 à 31)

```
data(airquality)
```

On peut trouver les informations concernant le jeu de données.

```
?airquality
```

# New York Air Quality Measurements

## Description

Daily air quality measurements in New York, May to September 1973.

## Usage

```
airquality
```

## Format

A data frame with 153 observations on 6 variables.

```
[,1] Ozone      numeric Ozone (ppb)
[,2] Solar.R    numeric Solar R (lang)
[,3] Wind       numeric Wind (mph)
[,4] Temp       numeric Temperature (degrees F)
[,5] Month      numeric Month (1--12)
[,6] Day        numeric Day of month (1--31)
```

# Chapitre 2

## L'importation de données : les packages readr et readxl

### 2.1 Présentation

- Le package readr : importation de données tabulaires (.csv, .tsv...). On peut trouver plus d'informations sur le package readr avec l'aide-mémoire associé : <https://readr.tidyverse.org/>
- Le package readxl : importation de données en format .xls ou .xlsx. On peut trouver plus d'informations sur le package readxl avec l'aide-mémoire associé : <https://readxl.tidyverse.org/>

Avant d'utiliser les fonctions liées à l'importation des données Excel, il faut au préalable charger la librairie associée. Ce n'est pas le cas pour les .csv ou .tsv car la librairie readr est incluse dans la librairie tidyverse.

```
library(readxl)
```

A noter qu'on peut importer des fichiers spss, stata et sas en utilisant le package haven (<https://haven.tidyverse.org/>).

### 2.2 Exemples

### 2.3 Exemple 1 : importation d'un fichier Excel

Nous utiliserons la fonction read\_excel() du package readxl.

```
exemple1 <- read_excel("exempleexcel.xlsx", na = "noobserv",  
                      col_names=TRUE)  
head(exemple1)
```

```
# A tibble: 5 x 4  
  colonne1 colonne2 colonne3 colonne4  
    <dbl>    <dbl>    <dbl> <chr>  
1      3.4        4        3 oui  
2        4        0        4 non
```

3	5	4.5	NA	oui
4	39	65	34	oui
5	10	100	43	non

## 2.4 Exemple 2 : importation d'un fichier .csv

Pour importer un fichier .csv, on peut utiliser la fonction `read_delim()` de la librairie `readr` incluse dans le package `tidyverse`.

```
# Séparateur : point-virgule
exemple2 <- read_delim("exempleexcel.csv", na = "noobserv",
                      delim = ";", show_col_types = FALSE)
head(exemple2)
```

```
# A tibble: 5 x 4
  colonne1 colonne2 colonne3 colonne4
  <dbl>    <dbl>    <dbl> <chr>
1      34         4         3 oui
2       4         0         4 non
3       5        45        NA oui
4      39        65        34 oui
5      10       100       43 non
```



# Chapitre 3

## Mise en forme et manipulation des données

### 3.1 Qu'est ce qu'un tibble ?

C'est la version moderne de la dataframe.

Pour construire un tableau de données, on utilise la fonction `tibble()` du package `tibble` (inclus dans le package `tidyverse`).

```
head(tibble(airquality))
```

```
# A tibble: 6 x 6
  Ozone Solar.R Wind Temp Month Day
  <int>   <int> <dbl> <int> <int> <int>
1    41     190  7.4    67     5    1
2    36     118   8     72     5    2
3    12     149 12.6    74     5    3
4    18     313 11.5    62     5    4
5    NA      NA 14.3    56     5    5
6    28      NA 14.9    66     5    6
```

Lorsqu'on importe un tableau de données avec `readr` or `readxl`, nous avons un tibble.

### 3.2 Quelques opérations élémentaires avec le package `dplyr`

#### 3.2.1 La commande pipe `%>%`

Effectuons la commande suivante.

```
airquality %>% head()
```

```
  Ozone Solar.R Wind Temp Month Day
1    41     190  7.4    67     5    1
2    36     118  8.0    72     5    2
3    12     149 12.6    74     5    3
4    18     313 11.5    62     5    4
5    NA      NA 14.3    56     5    5
6    28      NA 14.9    66     5    6
```

### 3.2.2 Création d'une nouvelle variable : `mutate()`

```
airquality %>% mutate(mois=month.name[Month]) %>% head()
```

	Ozone	Solar.R	Wind	Temp	Month	Day	mois
1	41	190	7.4	67	5	1	May
2	36	118	8.0	72	5	2	May
3	12	149	12.6	74	5	3	May
4	18	313	11.5	62	5	4	May
5	NA	NA	14.3	56	5	5	May
6	28	NA	14.9	66	5	6	May

On peut également ajouter une variable Year, les mesures ayant été faites en 1973.

```
airquality %>% mutate(Year=rep(1973,153)) %>% head()
```

	Ozone	Solar.R	Wind	Temp	Month	Day	Year
1	41	190	7.4	67	5	1	1973
2	36	118	8.0	72	5	2	1973
3	12	149	12.6	74	5	3	1973
4	18	313	11.5	62	5	4	1973
5	NA	NA	14.3	56	5	5	1973
6	28	NA	14.9	66	5	6	1973

### 3.2.3 Retour sur le pipe `%>%`

Cette commande est utilisée pour faire une succession d'actions.

```
ordre_mois<-c("May","June","July","August","September")
dataairquality <- airquality %>%
  as_tibble() %>%
  mutate(mois=month.name[Month]) %>%
  mutate(mois=factor(x=mois, levels=ordre_mois)) %>%
  mutate(Annee=rep(1973,153)) %>%
  drop_na()
# On enlève les lignes où il y a une donnée manquante.
```

### 3.2.4 Changer le nom d'une variable : `rename()`

```
(dataairquality <- dataairquality %>% rename(Mois=mois))
```

```
# A tibble: 111 x 8
```

	Ozone	Solar.R	Wind	Temp	Month	Day	Mois	Annee
	<int>	<int>	<dbl>	<int>	<int>	<int>	<fct>	<dbl>
1	41	190	7.4	67	5	1	May	1973
2	36	118	8	72	5	2	May	1973
3	12	149	12.6	74	5	3	May	1973
4	18	313	11.5	62	5	4	May	1973
5	23	299	8.6	65	5	7	May	1973
6	19	99	13.8	59	5	8	May	1973

```

7      8      19 20.1    61     5     9 May 1973
8     16     256 9.7     69     5    12 May 1973
9     11     290 9.2     66     5    13 May 1973
10    14     274 10.9    68     5    14 May 1973
# i 101 more rows

```

### 3.2.5 Sélectionner une ou plusieurs variable(s) : select()

```

# par le nom des variables
# Sélectionner la variable Ozone
dataairquality %>% select(Ozone)

```

```

# A tibble: 111 x 1
  Ozone
  <int>
1     41
2     36
3     12
4     18
5     23
6     19
7      8
8     16
9     11
10    14
# i 101 more rows

```

```

# Sélectionner les variables Ozone et Wind
dataairquality %>% select(Ozone, Wind)

```

```

# A tibble: 111 x 2
  Ozone Wind
  <int> <dbl>
1     41  7.4
2     36  8
3     12 12.6
4     18 11.5
5     23  8.6
6     19 13.8
7      8 20.1
8     16  9.7
9     11  9.2
10    14 10.9
# i 101 more rows

```

```

# par le numéro de colonne
dataairquality %>% select(c(1,3))

```

```

# A tibble: 111 x 2
  Ozone Wind

```

```

  <int> <dbl>
1    41    7.4
2    36     8
3    12   12.6
4    18   11.5
5    23    8.6
6    19   13.8
7     8   20.1
8    16    9.7
9    11    9.2
10   14   10.9
# i 101 more rows

```

```

# On peut également enlever une variable
dataairquality %>% select(-Month)

```

```

# A tibble: 111 x 7
  Ozone Solar.R Wind Temp Day Mois Annee
  <int>   <int> <dbl> <int> <int> <fct> <dbl>
1    41     190  7.4    67     1 May  1973
2    36     118  8      72     2 May  1973
3    12     149 12.6    74     3 May  1973
4    18     313 11.5    62     4 May  1973
5    23     299  8.6    65     7 May  1973
6    19      99 13.8    59     8 May  1973
7     8      19 20.1    61     9 May  1973
8    16     256  9.7    69    12 May  1973
9    11     290  9.2    66    13 May  1973
10   14     274 10.9    68    14 May  1973
# i 101 more rows

```

```

# Sélectionner les colonnes dont le nom commence par M
dataairquality %>% select(starts_with('M'))

```

```

# A tibble: 111 x 2
  Month Mois
  <int> <fct>
1     5 May
2     5 May
3     5 May
4     5 May
5     5 May
6     5 May
7     5 May
8     5 May
9     5 May
10    5 May
# i 101 more rows

```

```
# Sélectionner les colonnes dont le nom commence par M et finit par s
dataairquality %>% select(starts_with('M') & ends_with('s'))

# A tibble: 111 x 1
  Mois
  <fct>
1 May
2 May
3 May
4 May
5 May
6 May
7 May
8 May
9 May
10 May
# i 101 more rows
```

### 3.2.6 Création d'un sous-ensemble de données : filter()

```
dataairquality %>% filter(Mois=="May")

# A tibble: 24 x 8
  Ozone Solar.R Wind Temp Month Day Mois Annee
  <int>   <int> <dbl> <int> <int> <int> <fct> <dbl>
1    41     190  7.4   67    5    1 May  1973
2    36     118   8    72    5    2 May  1973
3    12     149 12.6   74    5    3 May  1973
4    18     313 11.5   62    5    4 May  1973
5    23     299  8.6   65    5    7 May  1973
6    19      99 13.8   59    5    8 May  1973
7     8      19 20.1   61    5    9 May  1973
8    16     256  9.7   69    5   12 May  1973
9    11     290  9.2   66    5   13 May  1973
10   14     274 10.9   68    5   14 May  1973
# i 14 more rows
```

```
dataairquality %>% filter(Wind >20)

# A tibble: 2 x 8
  Ozone Solar.R Wind Temp Month Day Mois Annee
  <int>   <int> <dbl> <int> <int> <int> <fct> <dbl>
1     8      19 20.1   61    5    9 May  1973
2    37     284 20.7   72    6   17 June  1973
```

```
dataairquality %>% filter(Wind <20 & Wind >10)

# A tibble: 51 x 8
  Ozone Solar.R Wind Temp Month Day Mois Annee
  <int>   <int> <dbl> <int> <int> <int> <fct> <dbl>
```

```

1    12    149  12.6   74    5    3 May   1973
2    18    313  11.5   62    5    4 May   1973
3    19     99  13.8   59    5    8 May   1973
4    14    274  10.9   68    5   14 May   1973
5    18     65  13.2   58    5   15 May   1973
6    14    334  11.5   64    5   16 May   1973
7    34    307  12     66    5   17 May   1973
8     6     78  18.4   57    5   18 May   1973
9    30    322  11.5   68    5   19 May   1973
10   11    320  16.6   73    5   22 May   1973
# i 41 more rows

```

```
dataairquality %>% filter(Wind >20 | Wind <3)
```

```
# A tibble: 4 x 8
```

	Ozone	Solar.R	Wind	Temp	Month	Day	Mois	Annee
	<int>	<int>	<dbl>	<int>	<int>	<int>	<fct>	<dbl>
1	8	19	20.1	61	5	9	May	1973
2	37	284	20.7	72	6	17	June	1973
3	118	225	2.3	94	8	29	August	1973
4	73	183	2.8	93	9	3	September	1973

### 3.2.7 Sélectionner des lignes en utilisant leur position : slice()

```
dataairquality %>% slice(c(1,3))
```

```
# A tibble: 2 x 8
```

	Ozone	Solar.R	Wind	Temp	Month	Day	Mois	Annee
	<int>	<int>	<dbl>	<int>	<int>	<int>	<fct>	<dbl>
1	41	190	7.4	67	5	1	May	1973
2	12	149	12.6	74	5	3	May	1973

### 3.2.8 Ordonner les lignes selon une (ou des) variable(s) : arrange()

```
dataairquality %>% arrange(Wind)
```

```
# A tibble: 111 x 8
```

	Ozone	Solar.R	Wind	Temp	Month	Day	Mois	Annee
	<int>	<int>	<dbl>	<int>	<int>	<int>	<fct>	<dbl>
1	118	225	2.3	94	8	29	August	1973
2	73	183	2.8	93	9	3	September	1973
3	168	238	3.4	81	8	25	August	1973
4	122	255	4	89	8	7	August	1973
5	135	269	4.1	84	7	1	July	1973
6	64	175	4.6	83	7	5	July	1973
7	91	189	4.6	93	9	4	September	1973
8	77	276	5.1	88	7	7	July	1973
9	79	187	5.1	87	7	19	July	1973
10	78	197	5.1	92	9	2	September	1973

```
# i 101 more rows
```

```
dataairquality %>% arrange(Wind, Temp)
```

```
# A tibble: 111 x 8
```

	Ozone	Solar.R	Wind	Temp	Month	Day	Mois	Annee
	<int>	<int>	<dbl>	<int>	<int>	<int>	<fct>	<dbl>
1	118	225	2.3	94	8	29	August	1973
2	73	183	2.8	93	9	3	September	1973
3	168	238	3.4	81	8	25	August	1973
4	122	255	4	89	8	7	August	1973
5	135	269	4.1	84	7	1	July	1973
6	64	175	4.6	83	7	5	July	1973
7	91	189	4.6	93	9	4	September	1973
8	79	187	5.1	87	7	19	July	1973
9	77	276	5.1	88	7	7	July	1973
10	78	197	5.1	92	9	2	September	1973

```
# i 101 more rows
```

```
# Par ordre décroissant
```

```
dataairquality %>% arrange(desc(Wind))
```

```
# A tibble: 111 x 8
```

	Ozone	Solar.R	Wind	Temp	Month	Day	Mois	Annee
	<int>	<int>	<dbl>	<int>	<int>	<int>	<fct>	<dbl>
1	37	284	20.7	72	6	17	June	1973
2	8	19	20.1	61	5	9	May	1973
3	6	78	18.4	57	5	18	May	1973
4	11	320	16.6	73	5	22	May	1973
5	14	20	16.6	63	9	25	September	1973
6	21	259	15.5	77	8	21	August	1973
7	32	92	15.5	84	9	6	September	1973
8	21	259	15.5	76	9	12	September	1973
9	45	252	14.9	81	5	29	May	1973
10	21	191	14.9	77	6	16	June	1973

```
# i 101 more rows
```

### 3.2.9 Calculer le nombre d'observations par groupe : count()

```
dataairquality %>% count(Mois)
```

```
# A tibble: 5 x 2
```

Mois	n
<fct>	<int>
1 May	24
2 June	9
3 July	26
4 August	23
5 September	29

### 3.2.10 Grouper : group\_by()

```
dataairquality %>% group_by(Mois)
```

# A tibble: 111 x 8

# Groups: Mois [5]

	Ozone	Solar.R	Wind	Temp	Month	Day	Mois	Annee
	<int>	<int>	<dbl>	<int>	<int>	<int>	<fct>	<dbl>
1	41	190	7.4	67	5	1	May	1973
2	36	118	8	72	5	2	May	1973
3	12	149	12.6	74	5	3	May	1973
4	18	313	11.5	62	5	4	May	1973
5	23	299	8.6	65	5	7	May	1973
6	19	99	13.8	59	5	8	May	1973
7	8	19	20.1	61	5	9	May	1973
8	16	256	9.7	69	5	12	May	1973
9	11	290	9.2	66	5	13	May	1973
10	14	274	10.9	68	5	14	May	1973

# i 101 more rows

Qu'observe-t-on ? Une nouvelle ligne a été ajoutée *Groups: Mois [5]*. Effectuons la ligne de commande suivante :

```
dataairquality %>% group_by(Mois) %>% slice(1)
```

# A tibble: 5 x 8

# Groups: Mois [5]

	Ozone	Solar.R	Wind	Temp	Month	Day	Mois	Annee
	<int>	<int>	<dbl>	<int>	<int>	<int>	<fct>	<dbl>
1	41	190	7.4	67	5	1	May	1973
2	29	127	9.7	82	6	7	June	1973
3	135	269	4.1	84	7	1	July	1973
4	39	83	6.9	81	8	1	August	1973
5	96	167	6.9	91	9	1	September	1973

Nous avons la première ligne du tableau de données pour chacun des mois.

### 3.2.11 Résumer des variables : summarise()

```
# Moyenne d'une variable
dataairquality %>% summarise(moyWind=mean(Wind))
```

# A tibble: 1 x 1

	moyWind
	<dbl>
1	9.94

```
# Calcul de la moyenne de chaque variable quantitative (sans les NA)
dataairquality %>% select(-Mois) %>% summarise_all(mean, na.rm=TRUE)
```

# A tibble: 1 x 7



```

Ozone Solar.R Wind Temp Month Day Annee
<dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 42.1 185. 9.94 77.8 7.22 15.9 1973

# calculer effectifs / mois
dataairquality %>% group_by(Mois) %>% summarize(n=n())

# A tibble: 5 x 2
  Mois      n
  <fct>    <int>
1 May      24
2 June      9
3 July     26
4 August   23
5 September 29

# Moyenne, variance, écart-type, médiane, min, max et IQR
dataairquality %>%
  summarise(Moy_wind = mean(Wind), var_wind=var(Wind),
            SD_wind=sd(Wind), mediane_Wind=median(Wind),
            min_wind=min(Wind), max_wind=max(Wind),
            IQR_wind=IQR(Wind))

# A tibble: 1 x 7
  Moy_wind var_wind SD_wind mediane_Wind min_wind max_wind IQR_wind
  <dbl>    <dbl>    <dbl>         <dbl>    <dbl>    <dbl>    <dbl>
1  9.94    12.7     3.56          9.7      2.3     20.7     4.1

```

### 3.2.12 Dégrouper : ungroup()

```

dataairquality %>% group_by(Mois) %>%
  count(Mois) %>% ungroup() %>%
  count(Mois)

```

```

# A tibble: 5 x 2
  Mois      n
  <fct>    <int>
1 May      1
2 June     1
3 July     1
4 August   1
5 September 1

```

### 3.2.13 Appliquer une fonction à chacun des groupes : group\_map()

```

dataairquality %>% group_by(Mois) %>%
  group_map(~quantile(.x$Ozone,
                      probs = c(0.25, 0.5, 0.75)))

```

```
[[1]]
 25%  50%  75%
11.0 18.0 32.5
```

```
[[2]]
 25%  50%  75%
 20  23  37
```

```
[[3]]
 25%  50%  75%
36.25 60.00 79.75
```

```
[[4]]
 25%  50%  75%
25.5 45.0 84.5
```

```
[[5]]
 25%  50%  75%
 16  23  36
```

### 3.2.14 Les lignes uniques d'un tableau de données : `distinct()`

```
dataairquality %>% distinct(Mois)
```

```
# A tibble: 5 x 1
  Mois
<fct>
1 May
2 June
3 July
4 August
5 September
```

```
dataairquality %>% distinct(Mois, Month)
```

```
# A tibble: 5 x 2
  Mois      Month
<fct>    <int>
1 May         5
2 June        6
3 July        7
4 August      8
5 September   9
```

### 3.3 Jointure de deux tableaux

```
tableau1 <- tibble(Mois=c("May", "August", "September"),
                   Value=c(3,7,50))

# inner_join
dataairquality %>% inner_join(tableau1)
```

Joining with `by = join\_by(Mois)`

# A tibble: 76 x 9

	Ozone	Solar.R	Wind	Temp	Month	Day	Mois	Annee	Value
	<int>	<int>	<dbl>	<int>	<int>	<int>	<chr>	<dbl>	<dbl>
1	41	190	7.4	67	5	1	May	1973	3
2	36	118	8	72	5	2	May	1973	3
3	12	149	12.6	74	5	3	May	1973	3
4	18	313	11.5	62	5	4	May	1973	3
5	23	299	8.6	65	5	7	May	1973	3
6	19	99	13.8	59	5	8	May	1973	3
7	8	19	20.1	61	5	9	May	1973	3
8	16	256	9.7	69	5	12	May	1973	3
9	11	290	9.2	66	5	13	May	1973	3
10	14	274	10.9	68	5	14	May	1973	3

# i 66 more rows

```
# left_join
dataairquality %>% left_join(tableau1)
```

Joining with `by = join\_by(Mois)`

# A tibble: 111 x 9

	Ozone	Solar.R	Wind	Temp	Month	Day	Mois	Annee	Value
	<int>	<int>	<dbl>	<int>	<int>	<int>	<chr>	<dbl>	<dbl>
1	41	190	7.4	67	5	1	May	1973	3
2	36	118	8	72	5	2	May	1973	3
3	12	149	12.6	74	5	3	May	1973	3
4	18	313	11.5	62	5	4	May	1973	3
5	23	299	8.6	65	5	7	May	1973	3
6	19	99	13.8	59	5	8	May	1973	3
7	8	19	20.1	61	5	9	May	1973	3
8	16	256	9.7	69	5	12	May	1973	3
9	11	290	9.2	66	5	13	May	1973	3
10	14	274	10.9	68	5	14	May	1973	3

# i 101 more rows

```
# anti_join
dataairquality %>% anti_join(tableau1)
```

Joining with `by = join\_by(Mois)`

# A tibble: 35 x 8

Ozone	Solar.R	Wind	Temp	Month	Day	Mois	Annee
<int>	<int>	<dbl>	<int>	<int>	<int>	<fct>	<dbl>

```

1    29    127   9.7   82    6    7 June 1973
2    71    291  13.8   90    6    9 June 1973
3    39    323  11.5   87    6   10 June 1973
4    23    148   8     82    6   13 June 1973
5    21    191  14.9   77    6   16 June 1973
6    37    284  20.7   72    6   17 June 1973
7    20     37   9.2   65    6   18 June 1973
8    12    120  11.5   73    6   19 June 1973
9    13    137  10.3   76    6   20 June 1973
10   135    269   4.1   84    7    1 July 1973
# i 25 more rows

```

## 3.4 Transformation de données : Le package tidyr

### 3.4.1 Enlever les données manquantes : drop\_na()

```
airquality %>% drop_na() %>% head()
```

```

Ozone Solar.R Wind Temp Month Day
1    41    190  7.4   67     5   1
2    36    118  8.0   72     5   2
3    12    149 12.6   74     5   3
4    18    313 11.5   62     5   4
5    23    299  8.6   65     5   7
6    19     99 13.8   59     5   8

```

### 3.4.2 Rassembler les colonnes en lignes : pivot\_longer()

On utilisera la fonction *pivot\_longer*. Par exemple, on souhaite créer une variable nommée *Variable* correspondant au nom des variables de mesures à laquelle on va associer une autre variable nommée *Valeur* donnant la valeur de chaque mesure. De cette manière, on diminue le nombre de colonnes et on augmente le nombre de lignes.

```

pivot_longer(dataairquality, cols=1:4, names_to="Variable",
              values_to="Valeur")

```

```

# A tibble: 444 x 6
  Month Day Mois  Annee Variable Valeur
  <int> <int> <fct> <dbl> <chr>    <dbl>
1     5     1 May   1973 Ozone     41
2     5     1 May   1973 Solar.R  190
3     5     1 May   1973 Wind      7.4
4     5     1 May   1973 Temp      67
5     5     2 May   1973 Ozone     36
6     5     2 May   1973 Solar.R  118
7     5     2 May   1973 Wind      8
8     5     2 May   1973 Temp      72
9     5     3 May   1973 Ozone     12
10    5     3 May   1973 Solar.R  149

```

```
# i 434 more rows
```

### 3.4.3 Répartir les lignes en colonnes : spread()

```
dataairquality %>% spread(Mois, Day)
```

```
# A tibble: 111 x 11
```

	Ozone	Solar.R	Wind	Temp	Month	Annee	May	June	July	August	September
	<int>	<int>	<dbl>	<int>	<int>	<dbl>	<int>	<int>	<int>	<int>	<int>
1	1	8	9.7	59	5	1973	21	NA	NA	NA	NA
2	4	25	9.7	61	5	1973	23	NA	NA	NA	NA
3	6	78	18.4	57	5	1973	18	NA	NA	NA	NA
4	7	48	14.3	80	7	1973	NA	NA	15	NA	NA
5	7	49	10.3	69	9	1973	NA	NA	NA	NA	24
6	8	19	20.1	61	5	1973	9	NA	NA	NA	NA
7	9	24	10.9	71	9	1973	NA	NA	NA	NA	14
8	9	24	13.8	81	8	1973	NA	NA	NA	2	NA
9	9	36	14.3	72	8	1973	NA	NA	NA	22	NA
10	10	264	14.3	73	7	1973	NA	NA	12	NA	NA

```
# i 101 more rows
```

### 3.4.4 Créer des listes de tableaux de données : nest()

```
dataairquality %>% group_by(Mois) %>% nest()
```

```
# A tibble: 5 x 2
```

```
# Groups:   Mois [5]
```

Mois	data
<fct>	<list>
1 May	<tibble [24 x 7]>
2 June	<tibble [9 x 7]>
3 July	<tibble [26 x 7]>
4 August	<tibble [23 x 7]>
5 September	<tibble [29 x 7]>

### 3.4.5 Aplatir en colonnes régulières : unnest()

```
dataairquality %>% group_by(Mois) %>%
  nest() %>%
  unnest(c(1,2))
```

```
# A tibble: 111 x 8
```

```
# Groups:   Mois [5]
```

Mois	Ozone	Solar.R	Wind	Temp	Month	Day	Annee
<fct>	<int>	<int>	<dbl>	<int>	<int>	<int>	<dbl>
1 May	41	190	7.4	67	5	1	1973
2 May	36	118	8	72	5	2	1973
3 May	12	149	12.6	74	5	3	1973
4 May	18	313	11.5	62	5	4	1973

5 May	23	299	8.6	65	5	7	1973
6 May	19	99	13.8	59	5	8	1973
7 May	8	19	20.1	61	5	9	1973
8 May	16	256	9.7	69	5	12	1973
9 May	11	290	9.2	66	5	13	1973
10 May	14	274	10.9	68	5	14	1973

# i 101 more rows

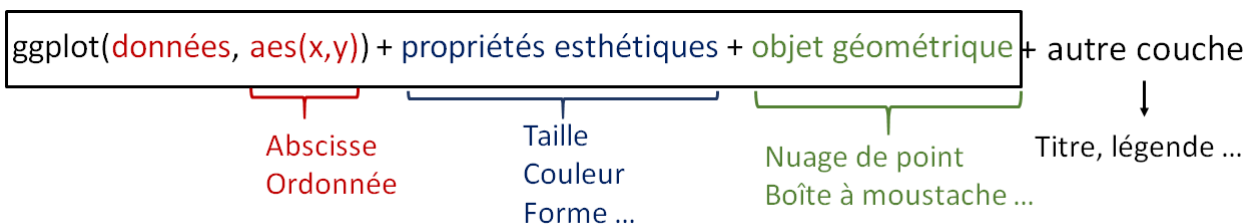
# Chapitre 4

## Les graphiques avec le package ggplot2

### 4.1 Le principe

Vous pouvez aller sur le site <https://ggplot2.tidyverse.org>. Un aide-mémoire est disponible.

- Package développé par H. Wickham
- Grammaire des graphiques s'appuyant sur le jeu de données, le système de coordonnées, l'objet géométrique.
- Principale fonction : `ggplot()`

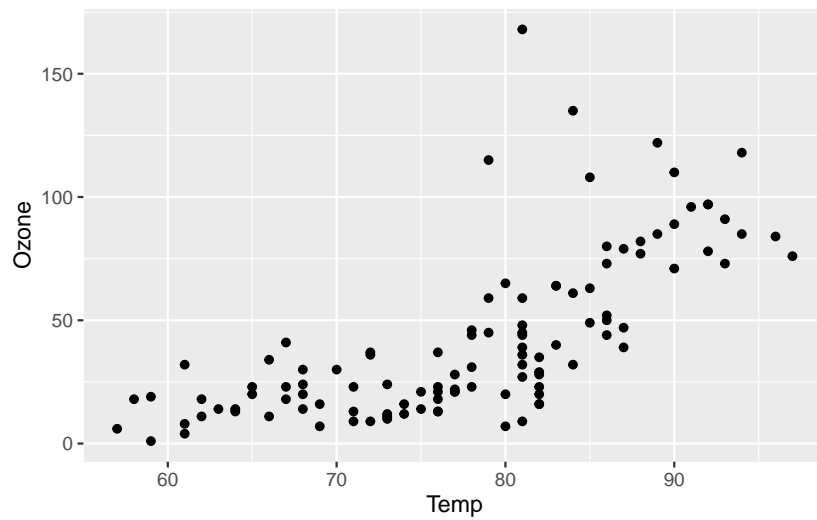


### 4.2 Exemple d'un nuage de points

#### 4.2.1 Les commandes élémentaires

Reprenons le jeu de données `dataairquality`. Traçons un premier nuage de points avec en abscisse la température et en ordonnée le taux d'ozone.

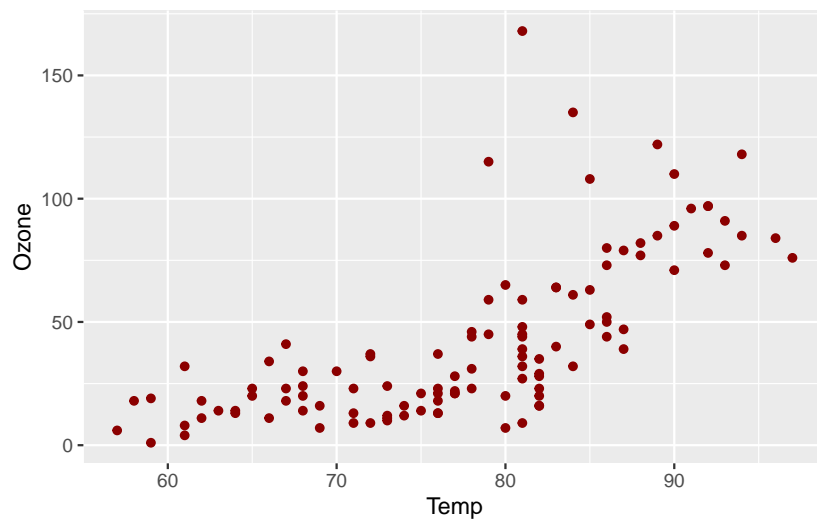
```
ggplot(dataairquality, aes(x=Temp, y=Ozone)) + geom_point()
```



Exécutez la commande `ggplot(dataairquality, aes(x=Temp, y=Ozone))`. Qu'observez-vous ?

Traçons les points en rouge :

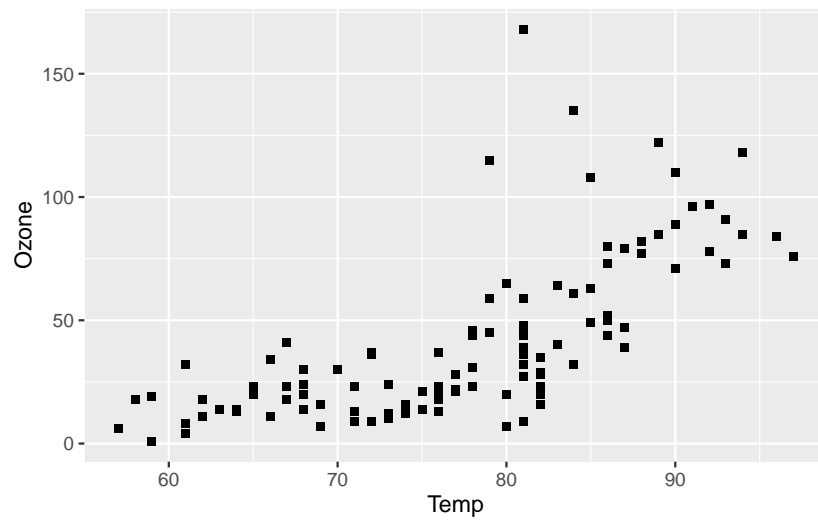
```
ggplot(dataairquality, aes(x=Temp, y=Ozone)) +  
  geom_point(colour="darkred")
```



Changeons la forme des points en carré.

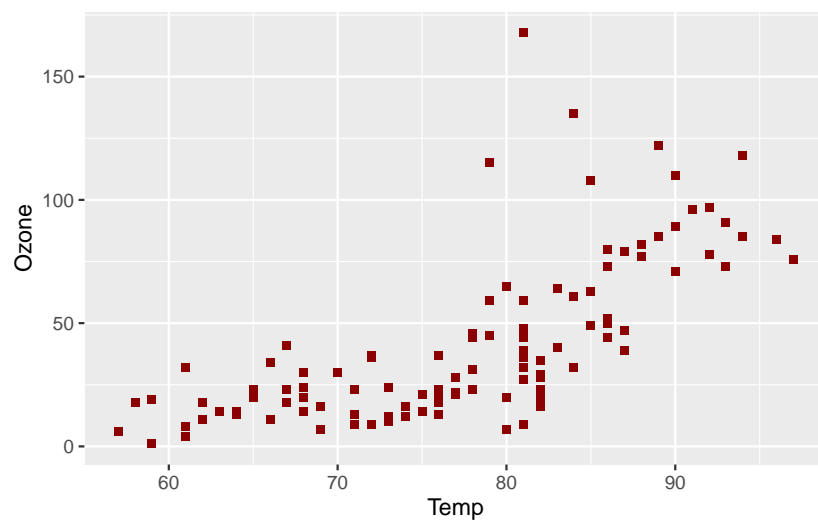
```
ggplot(dataairquality, aes(x=Temp, y=Ozone)) +  
  geom_point(pch=15)
```





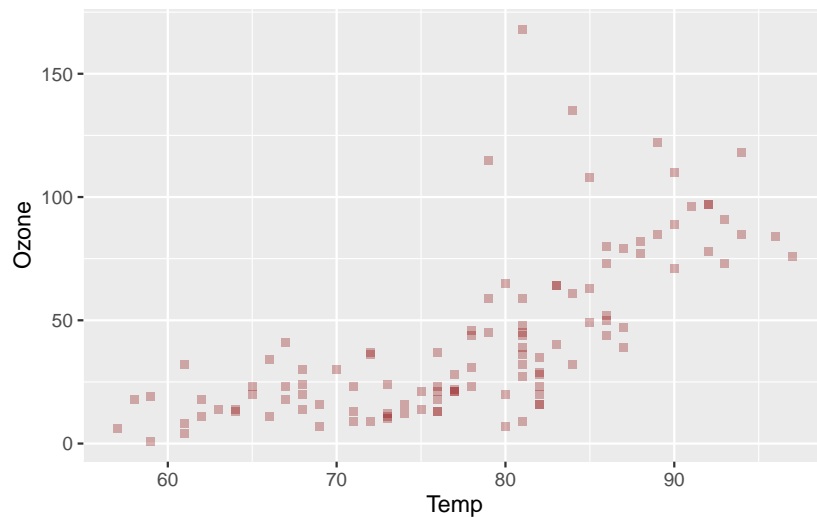
Si on souhaite tracer tous les carrés en rouge, il faut inclure toutes les options.

```
ggplot(dataairquality, aes(x=Temp, y=Ozone)) +  
  geom_point(colour="darkred", pch=15)
```



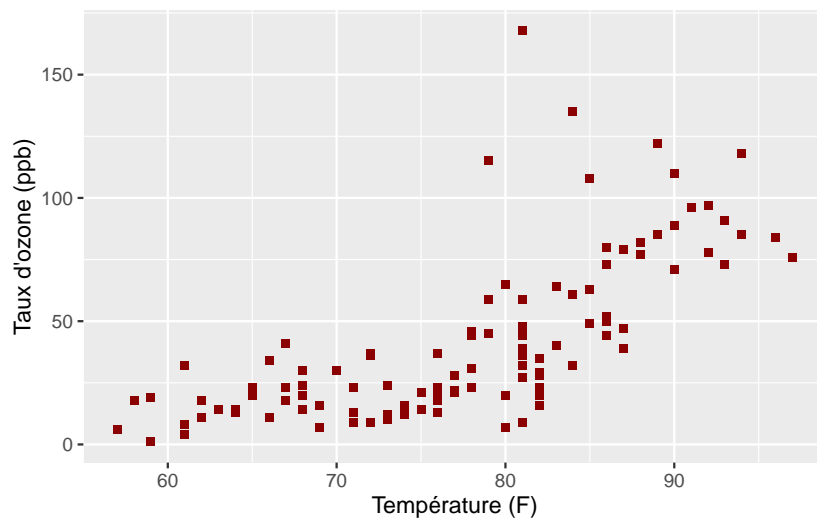
Si on souhaite modifier la transparence des points, on peut utiliser la fonction `alpha`.

```
ggplot(dataairquality, aes(x=Temp, y=Ozone)) +  
  geom_point(colour="darkred", alpha=0.3, pch=15)
```



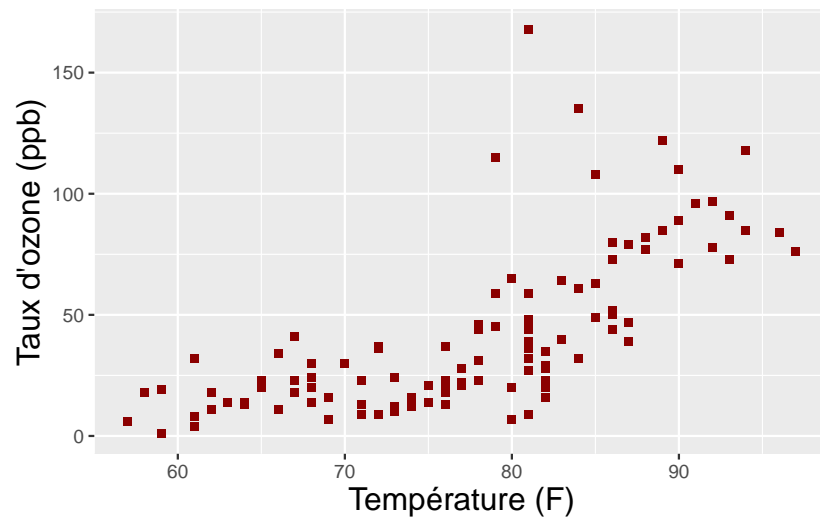
On peut modifier le titre de l'axe des abscisses avec `xlab` en ajoutant une couche. Pour l'axe des ordonnées, on utilisera `ylab`.

```
ggplot(dataairquality, aes(x=Temp, y=Ozone)) +
  geom_point(colour="darkred", pch=15) +
  xlab("Température (F)") + ylab("Taux d'ozone (ppb)")
```



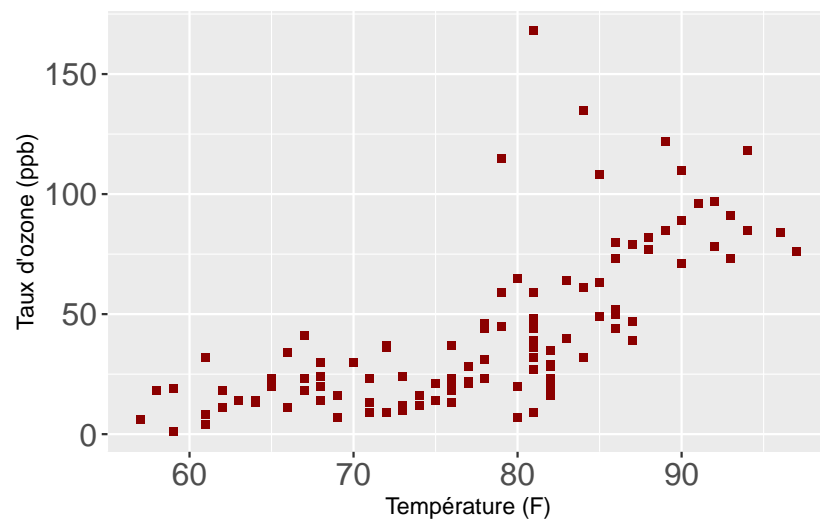
On peut changer la taille du titre des axes.

```
ggplot(dataairquality, aes(x=Temp, y=Ozone)) +
  geom_point(colour="darkred", pch=15) +
  xlab("Température (F)") + ylab("Taux d'ozone (ppb)") +
  theme(axis.title.x = element_text(size = 15),
        axis.title.y = element_text(size = 15))
```



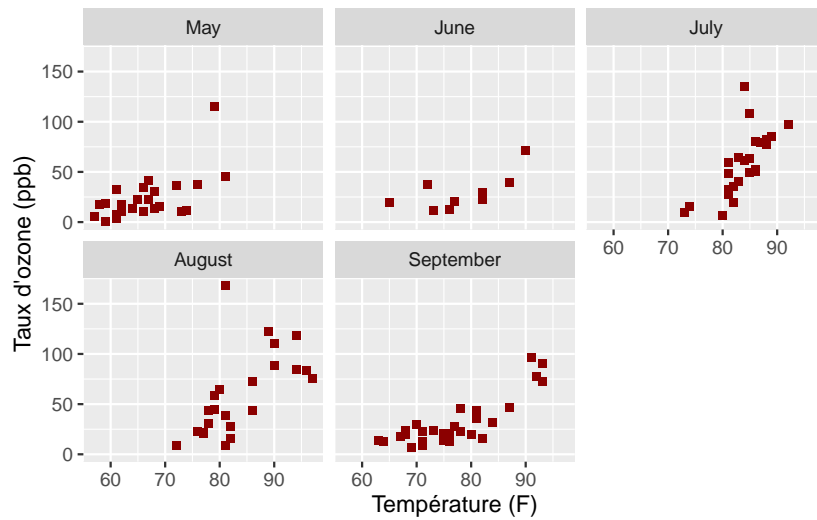
On peut changer la taille des étiquettes des graduations des axes.

```
ggplot(dataairquality, aes(x=Temp, y=Ozone)) +
  geom_point(colour="darkred", pch=15) +
  xlab("Température (F)") + ylab("Taux d'ozone (ppb)") +
  theme(axis.text = element_text(size = 15))
```



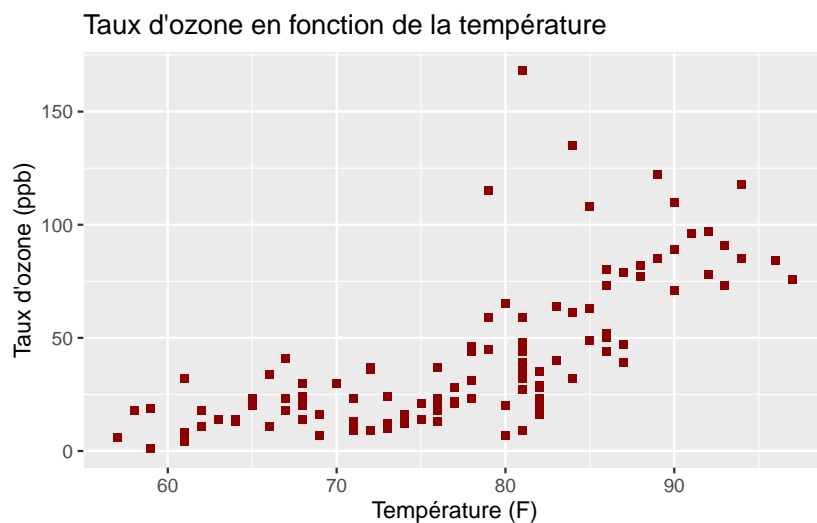
Si on souhaite effectuer un graphique pour chaque mois, on peut utiliser `facet_wrap()`.

```
ggplot(dataairquality, aes(x=Temp, y=Ozone)) +
  geom_point(colour="darkred", pch=15) +
  xlab("Température (F)") + ylab("Taux d'ozone (ppb)") +
  facet_wrap(~Mois)
```



On peut également ajouter un titre au graphique en utilisant `ggtitle`.

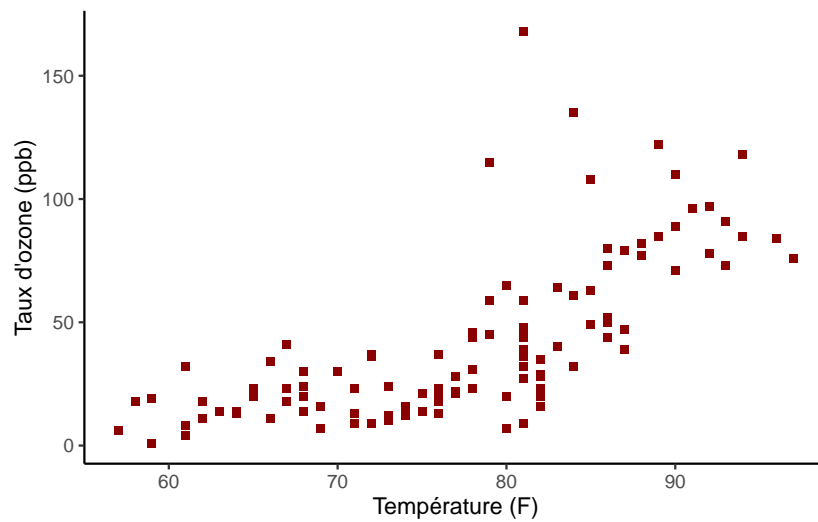
```
ggplot(dataairquality, aes(x=Temp, y=Ozone)) +
  geom_point(colour="darkred", pch=15) +
  xlab("Température (F)") + ylab("Taux d'ozone (ppb)") +
  ggtitle("Taux d'ozone en fonction de la température")
```



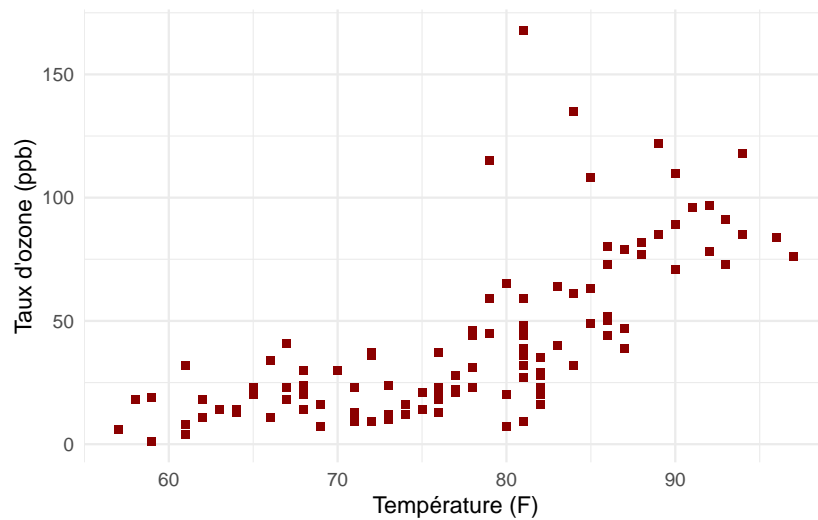
## 4.2.2 Theme

Si on regarde le graphique, on constate qu'un thème notamment avec un fond gris a été mis par défaut. On peut modifier le thème soit en utilisant un thème déjà implémenté soit en le modifiant manuellement. D'autres packages existent pour utiliser des fonds déjà implémentés comme par exemple le package `ggthemes`.

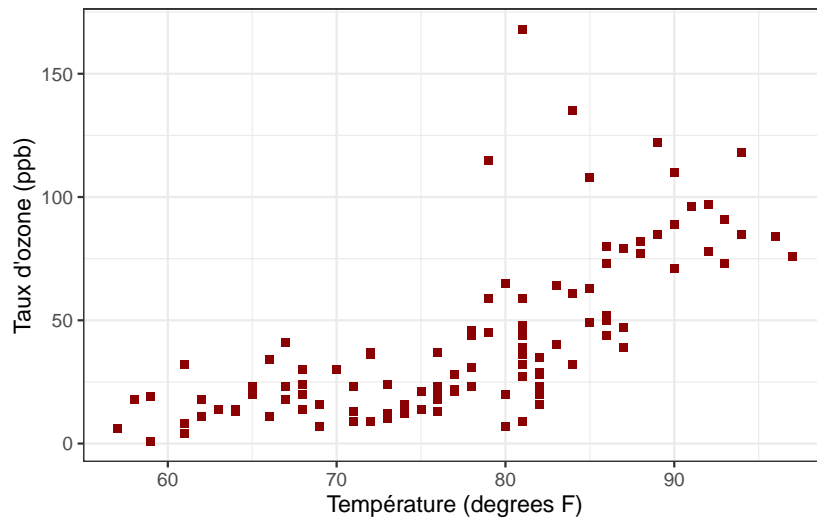
```
ggplot(dataairquality, aes(x=Temp, y=Ozone)) +
  geom_point(colour="darkred", pch=15) +
  xlab("Température (F)") + ylab("Taux d'ozone (ppb)") +
  theme_classic()
```



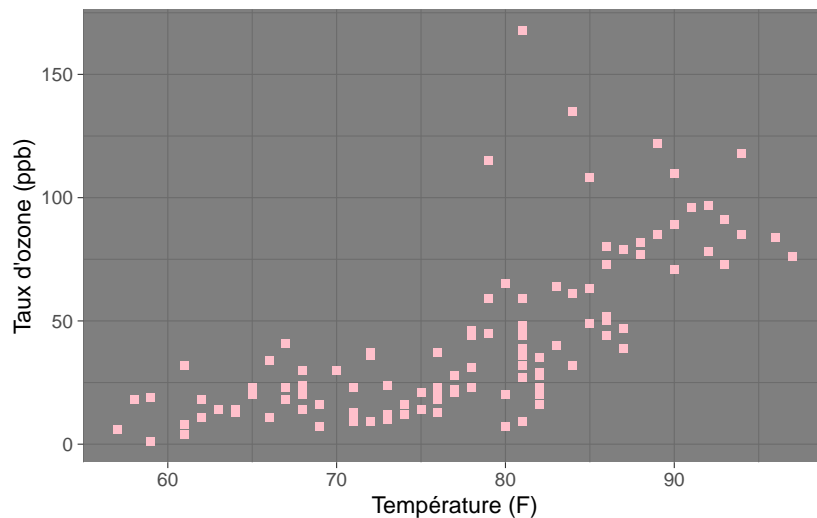
```
ggplot(dataairquality, aes(x=Temp, y=Ozone)) +
  geom_point(colour="darkred", pch=15) +
  xlab("Température (F)") + ylab("Taux d'ozone (ppb)") +
  theme_minimal()
```



```
ggplot(dataairquality, aes(x=Temp, y=Ozone)) +
  geom_point(colour="darkred", pch=15) +
  xlab("Température (degrees F)") + ylab("Taux d'ozone (ppb)") +
  theme_bw()
```

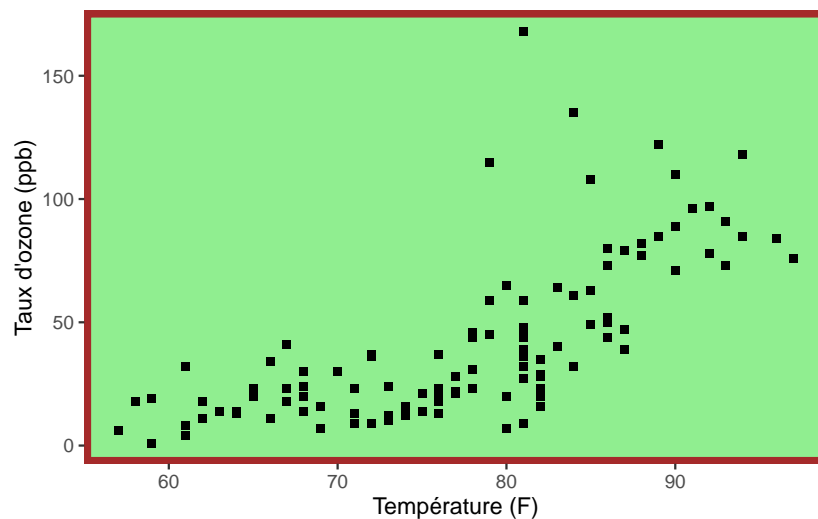


```
ggplot(dataairquality, aes(x=Temp, y=Ozone)) +
  geom_point(colour="pink", pch=15) +
  xlab("Température (F)") + ylab("Taux d'ozone (ppb)") +
  theme_dark()
```



On peut également modifier le thème manuellement :

```
ggplot(dataairquality, aes(x=Temp, y=Ozone)) +
  geom_point(colour="black", pch=15) +
  xlab("Température (F)") + ylab("Taux d'ozone (ppb)") +
  theme(panel.background = element_rect(fill="lightgreen"),
        panel.border = element_rect(colour="brown", fill=NA, linewidth=3),
        panel.grid.minor = element_blank(),
        panel.grid.major = element_blank())
```



De nombreuses options existent dans la fonction `theme()`.

R: Modify components of a theme ▾
Find in Topic

## Modify components of a theme

**Description**

Themes are a powerful way to customize the non-data components of your plots: i.e. titles, labels, fonts, background, gridlines, and legends. Themes can be used to give plots a consistent customized look. Modify a single plot's theme using `theme()`; see `theme_update()` if you want modify the active theme, to affect all subsequent plots. Use the themes available in [complete themes](#) if you would like to use a complete theme such as `theme_bw()`, `theme_minimal()`, and more. Theme elements are documented together according to inheritance, read more about theme inheritance below.

**Usage**

```
theme(
  line,
  rect,
  text,
  title,
  aspect.ratio,
  axis.title,
```

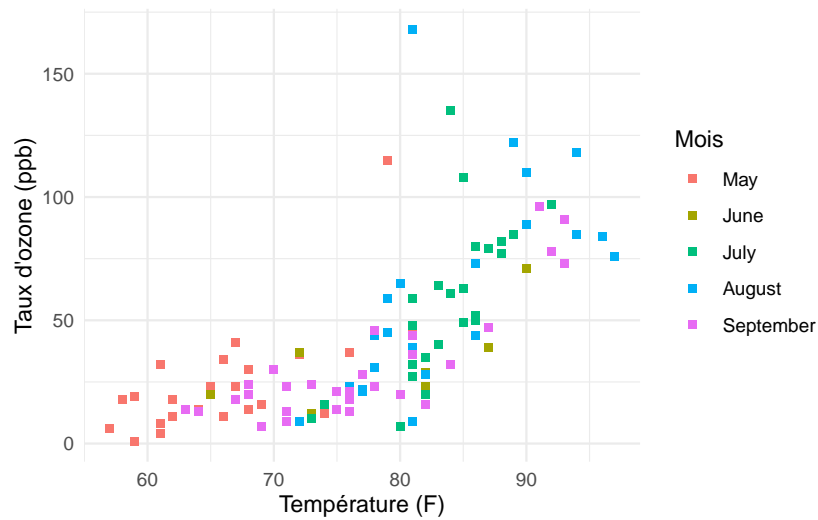
Vous pouvez consulter le lien suivant pour avoir les options modifiables dans `theme()` : <https://ggplot2.tidyverse.org/reference/theme.html>

Vous pouvez également utiliser le package `ggThemeAssist` (à installer auparavant) et la fonction `ggThemeAssistGadget(nom de l'objet ggplot2 à tracer)`.

### 4.2.3 Les couleurs

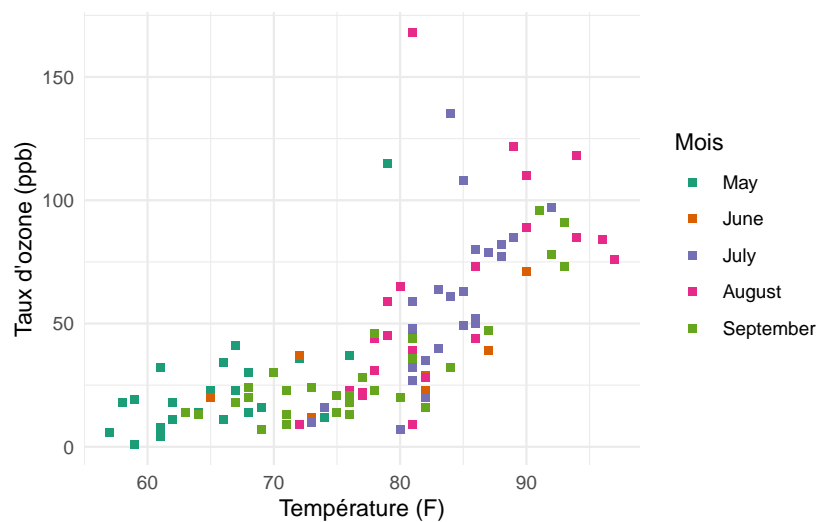
On souhaite à présent colorier chaque point en fonction du mois qui est une variable qualitative.

```
ggplot(dataairquality, aes(x=Temp, y=Ozone)) +
  geom_point(pch=15, aes(colour=Mois)) +
  xlab("Température (F)") + ylab("Taux d'ozone (ppb)") +
  theme_minimal()
```



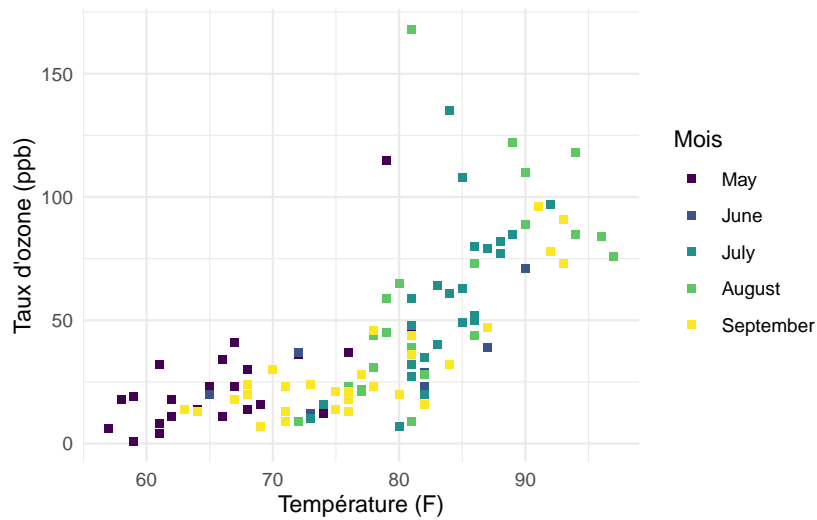
On peut également utiliser les palettes de couleur.

```
ggplot(dataairquality, aes(x=Temp, y=Ozone)) +
  geom_point(pch=15, aes(colour=Mois))+
  xlab("Température (F)") + ylab("Taux d'ozone (ppb)") +
  theme_minimal() +
  scale_color_brewer(palette="Dark2")
```



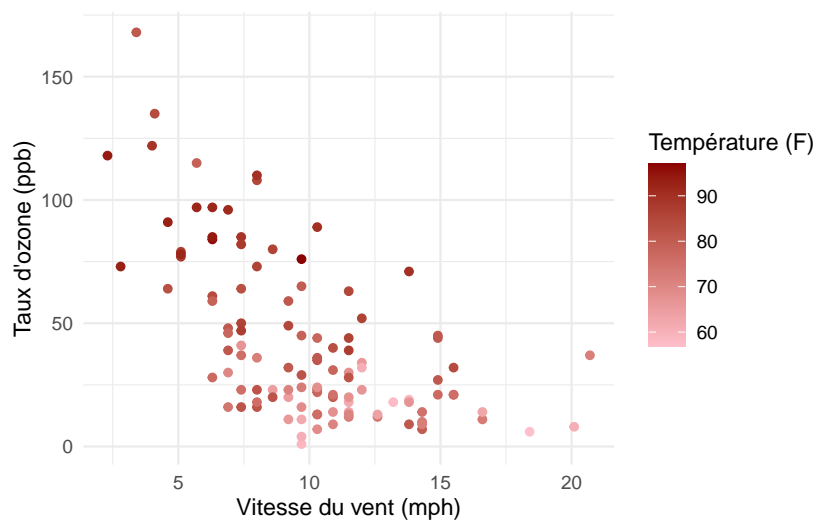
```
ggplot(dataairquality, aes(x=Temp, y=Ozone)) +
  geom_point(pch=15, aes(colour=Mois))+
  xlab("Température (F)") + ylab("Taux d'ozone (ppb)") +
  theme_minimal() +
  scale_color_viridis_d()
```





Traçons le taux d'ozone en fonction de la vitesse du vent en coloriant les points selon l'intensité de la température (variable quantitative).

```
ggplot(dataairquality, aes(x=Wind, y=Ozone)) +
  geom_point(aes(colour=Temp)) +
  scale_colour_gradient(low="pink", high="darkred") +
  xlab("Vitesse du vent (mph)") + ylab("Taux d'ozone (ppb)") +
  theme_minimal() +
  labs(colour="Température (F)")
```




Si vous souhaitez utiliser une palette de couleurs pour des articles scientifiques, un package est particulièrement intéressant : le package `ggsci` .


```
library(ggsci)
```

```
ggplot(dataairquality, aes(x=Temp, y=Ozone)) +
  geom_point(pch=15, aes(colour=Mois))+
  xlab("Température (F)") + ylab("Taux d'ozone (ppb)") +
```

# Scientific Journal and Sci-Fi Themed Color Palettes for 'ggplot2'




---



## Documentation for package 'ggsci' version 2.9

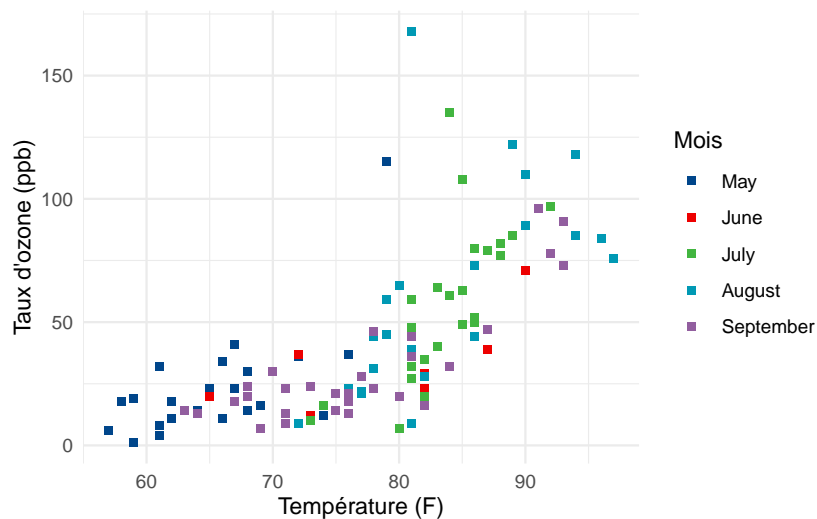
- [DESCRIPTION file.](#)
- [User guides, package vignettes and other documentation.](#)

## Help Pages

<a href="#">ggsci-package</a>	Scientific Journal and Sci-Fi Themed Color Palettes for 'ggplot2'
<a href="#">pal_aaas</a>	AAAS Journal Color Palettes
<a href="#">pal_d3</a>	D3.js Color Palettes
<a href="#">pal_futurama</a>	The Futurama Color Palettes
<a href="#">pal_gsea</a>	The GSEA GenePattern Color Palettes
<a href="#">pal_igv</a>	Integrative Genomics Viewer (IGV) Color Palettes
<a href="#">pal_jama</a>	Journal of the American Medical Association Color Palettes

Figure 4.1: Page d'aide du package ggsci

```
theme_minimal() +
scale_color_lancet()
```



Une librairie `khroma` a été créé pour les personnes daltoniennes permettant ainsi que les graphiques soient accessibles au plus grand nombre de lecteurs.

```
library(khroma)
```

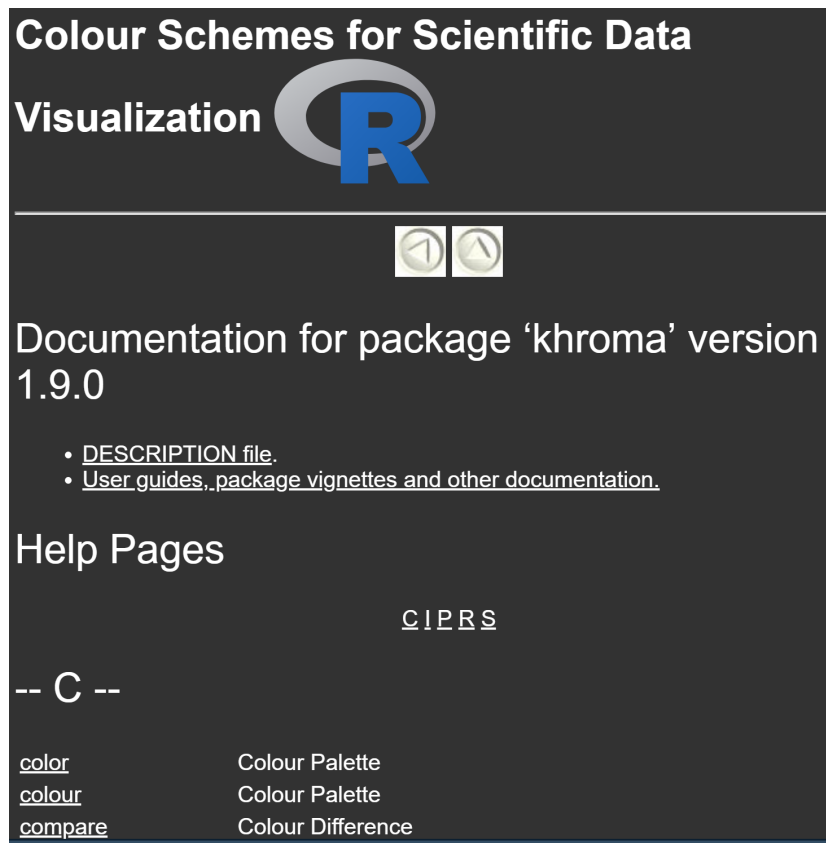
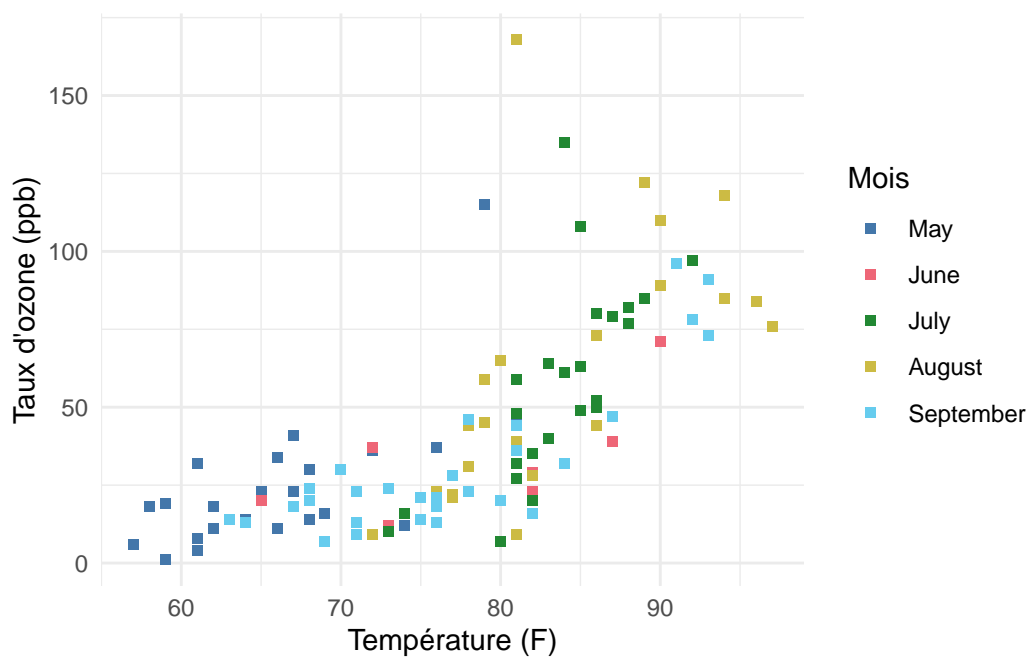


Figure 4.2: Page d'aide du package khroma

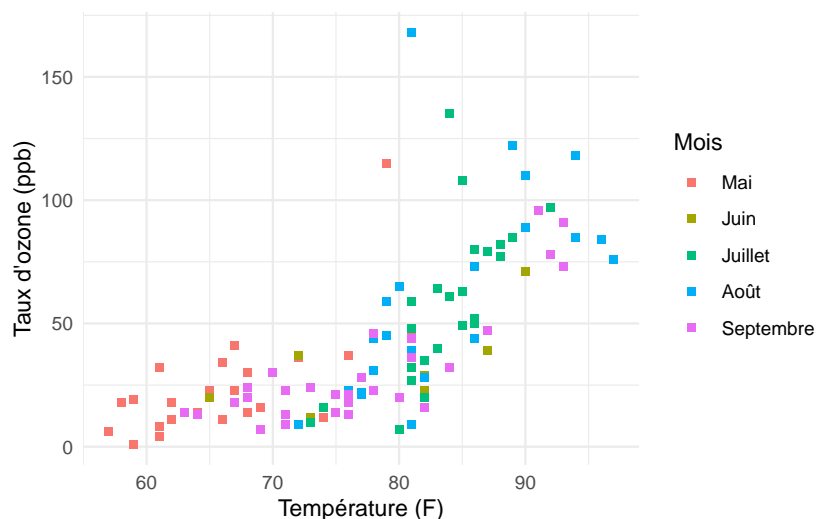
```
ggplot(dataairquality, aes(x=Temp, y=Ozone)) +
  geom_point(pch=15, aes(colour=Mois))+
  xlab("Température (F)") + ylab("Taux d'ozone (ppb)") +
  theme_minimal() +
  scale_colour_bright()
```



### 4.2.4 La légende

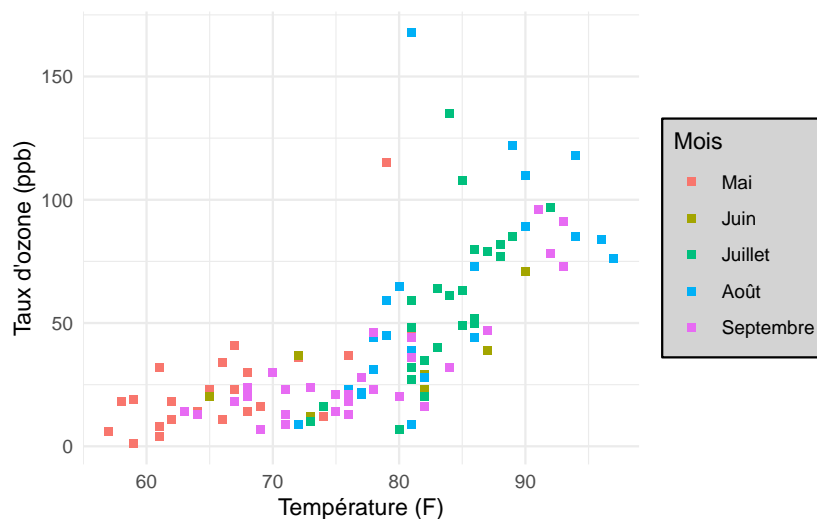
On peut modifier le texte de la légende.

```
ggplot(dataairquality, aes(x=Temp, y=Ozone)) +
  geom_point(pch=15, aes(colour=Mois))+
  xlab("Température (F)") + ylab("Taux d'ozone (ppb)") +
  theme_minimal() +
  scale_colour_discrete(labels=
    c("Mai", "Juin", "Juillet", "Août", "Septembre"))
```



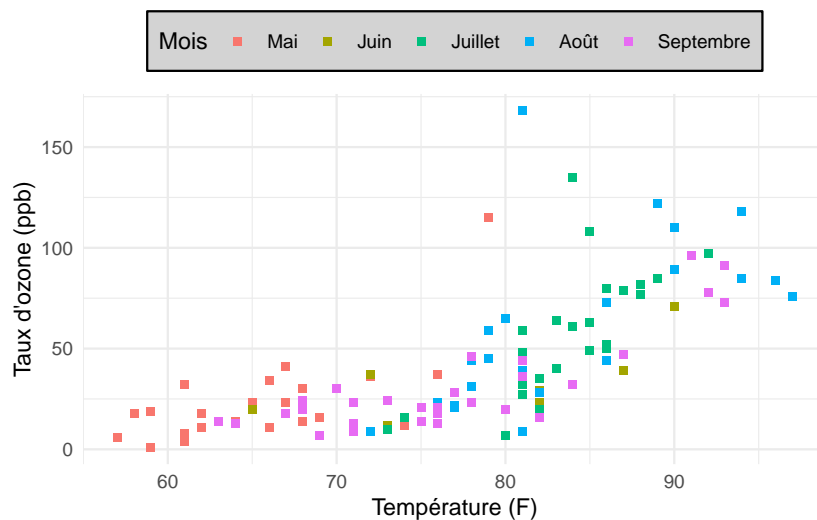
On peut également changer l'aspect de la légende.

```
ggplot(dataairquality, aes(x=Temp, y=Ozone)) +
  geom_point(pch=15, aes(colour=Mois))+
  xlab("Température (F)") + ylab("Taux d'ozone (ppb)") +
  theme_minimal() +
  scale_colour_discrete(labels=
    c("Mai", "Juin", "Juillet", "Août", "Septembre")) +
  theme(legend.background = element_rect(fill="lightgrey"))
```

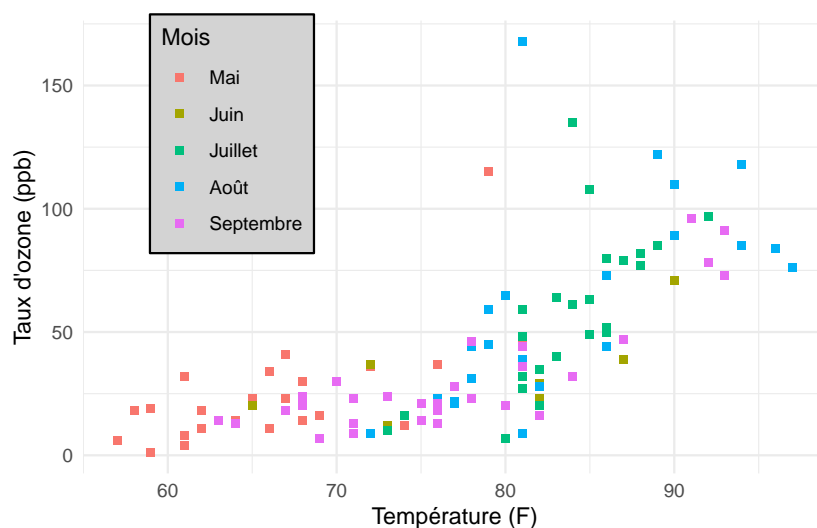


On peut changer la position de la légende :

```
ggplot(dataairquality, aes(x=Temp, y=Ozone)) +
  geom_point(pch=15, aes(colour=Mois))+
  xlab("Température (F)") + ylab("Taux d'ozone (ppb)") +
  theme_minimal() +
  scale_colour_discrete(labels=
    c("Mai", "Juin", "Juillet", "Août", "Septembre")) +
  theme(legend.position = "top",
    legend.background = element_rect(fill="lightgrey"))
```



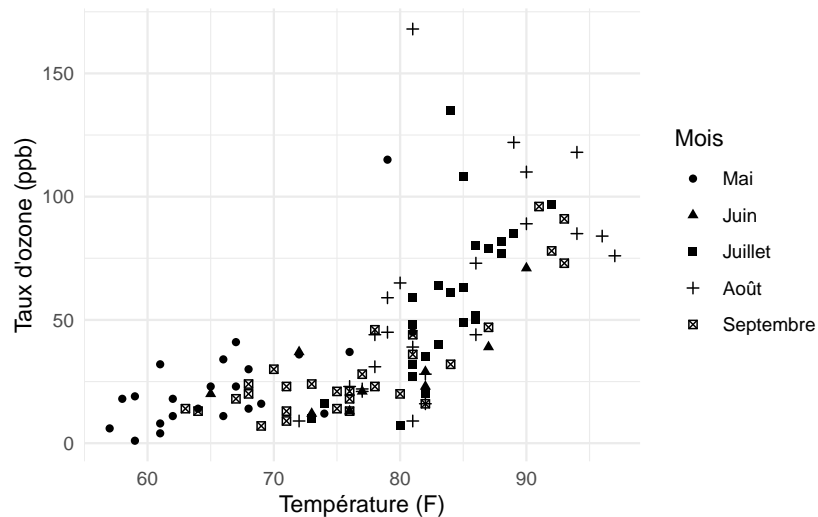
```
ggplot(dataairquality, aes(x=Temp, y=Ozone)) +
  geom_point(pch=15, aes(colour=Mois))+
  xlab("Température (F)") + ylab("Taux d'ozone (ppb)") +
  theme_minimal() +
  scale_colour_discrete(labels=
    c("Mai", "Juin", "Juillet", "Août", "Septembre")) +
  theme(legend.position = c(.2,.75),
    legend.background = element_rect(fill="lightgrey"))
```



### 4.2.5 La forme

On souhaite à présent changer la forme du point selon le mois.

```
ggplot(dataairquality, aes(x=Temp, y=Ozone)) +
  geom_point(aes(shape=Mois))+
  xlab("Température (F)") + ylab("Taux d'ozone (ppb)") +
  theme_minimal() +
  scale_shape_discrete(labels =
    c("Mai", "Juin", "Juillet", "Août", "Septembre"))
```



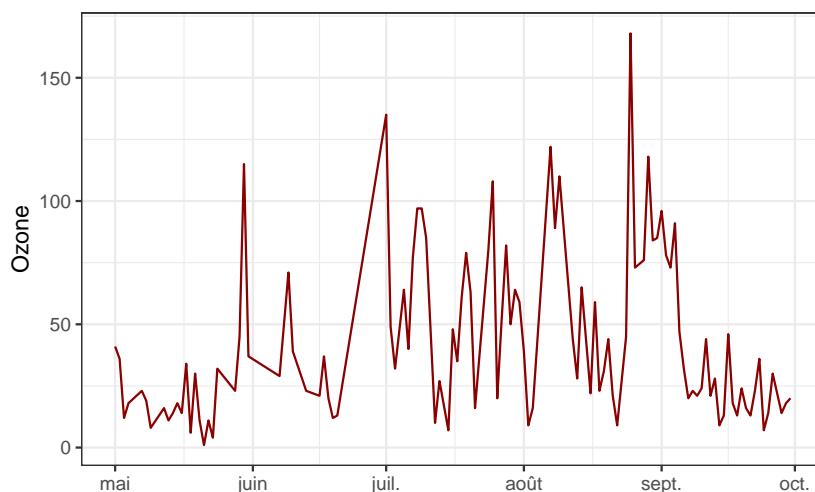
### 4.2.6 Représenter une variable temporelle

Tout d'abord, créons une variable date. Il faut au préalable charger le package lubridate.

```
library(lubridate)
dataairquality2 <- dataairquality %>%
  mutate(Date = dmy(paste(Day, Month, Annee, sep = "/")))
```

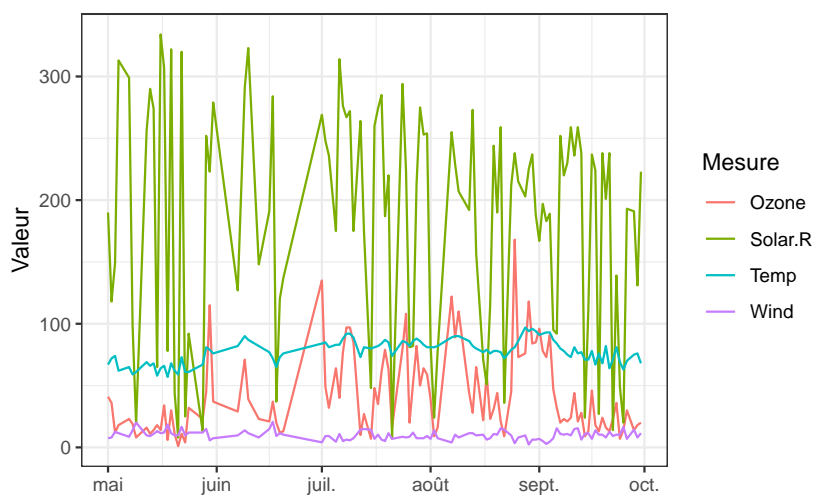
Traçons le taux d'ozone en fonction du mois.

```
ggplot(dataairquality2, aes(x=Date, y=Ozone)) +
  geom_line(colour="darkred") +
  xlab("") + theme_bw()
```



Si on souhaite représenter toutes les mesures sur le même graphique, il faut se ramener au chapitre précédent sur la manipulation des données.

```
dataairquality_allmes <- pivot_longer(dataairquality2, cols=1:4,
                                       names_to = "Mesure",
                                       values_to = "Valeur")
ggplot(dataairquality_allmes, aes(x=Date, y=Valeur)) +
  geom_line(aes(colour=Mesure)) +
  xlab("") + theme_bw()
```

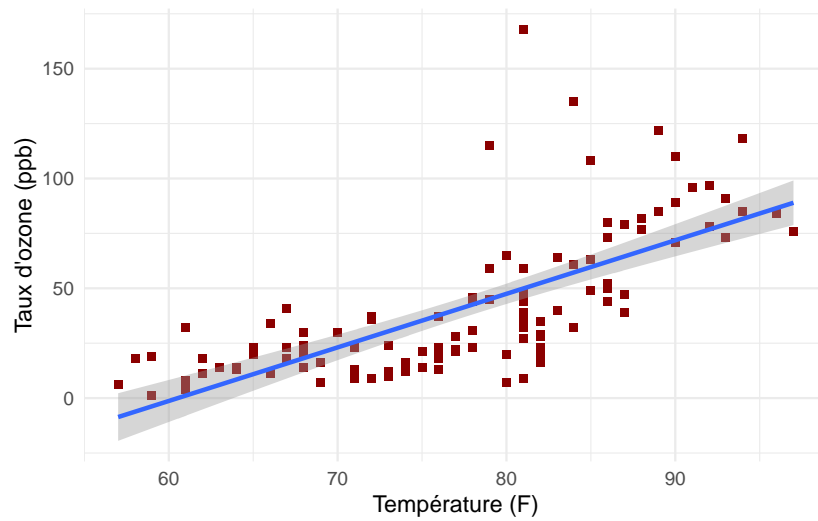


## 4.3 Ajouter des éléments statistiques

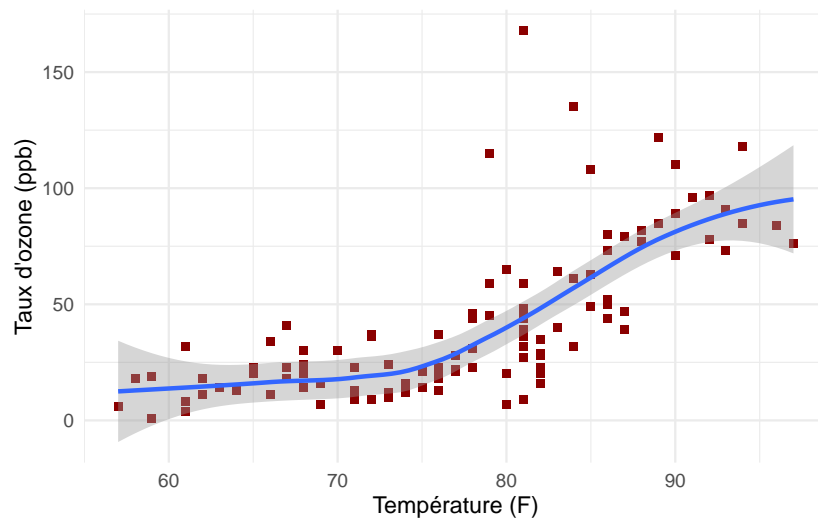
### 4.3.1 Ajouter une droite de régression

```
ggplot(dataairquality, aes(x=Temp, y=Ozone)) +
  geom_point(colour="darkred", pch=15) +
  geom_smooth(method="lm", formula=y~x) +
  xlab("Température (F)") + ylab("Taux d'ozone (ppb)") +
```

```
theme_minimal()
```

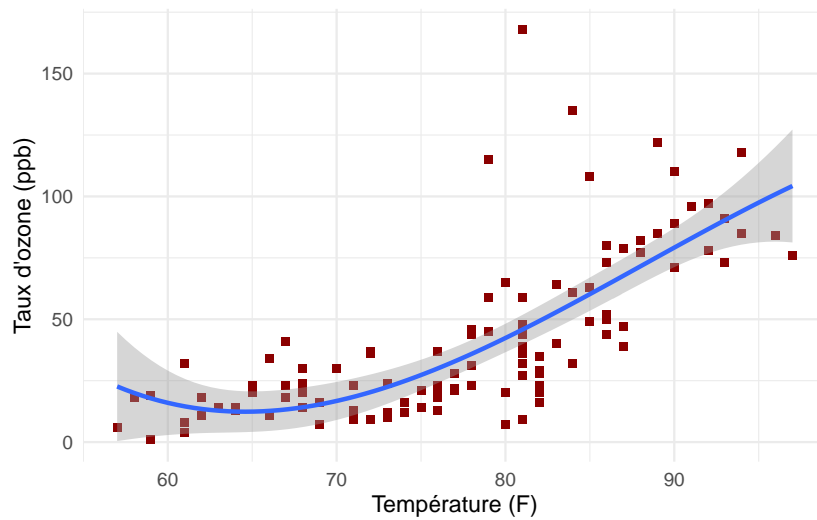


```
ggplot(dataairquality, aes(x=Temp, y=Ozone)) +  
  geom_point(colour="darkred", pch=15) +  
  geom_smooth(method="loess", formula=y~x) +  
  xlab("Température (F)") + ylab("Taux d'ozone (ppb)") +  
  theme_minimal()
```

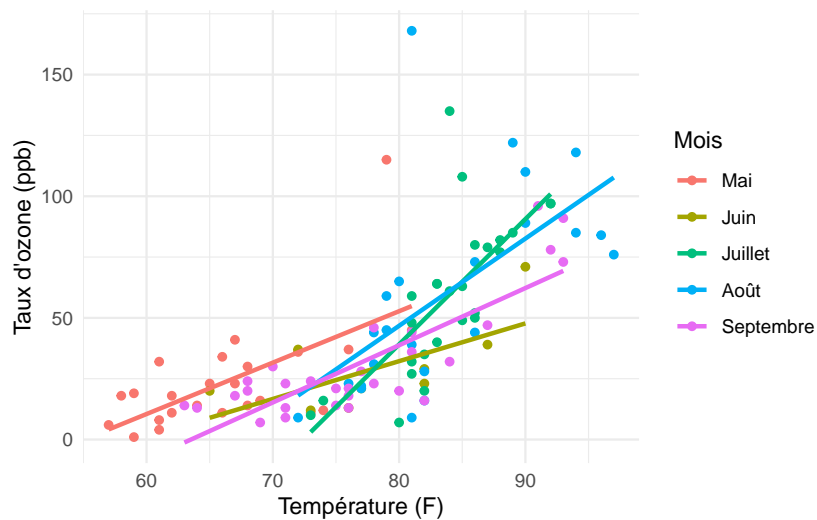


```
ggplot(dataairquality, aes(x=Temp, y=Ozone)) +  
  geom_point(colour="darkred", pch=15) +  
  geom_smooth(method="lm", formula=y~poly(x,3)) +  
  xlab("Température (F)") + ylab("Taux d'ozone (ppb)") +  
  theme_minimal()
```





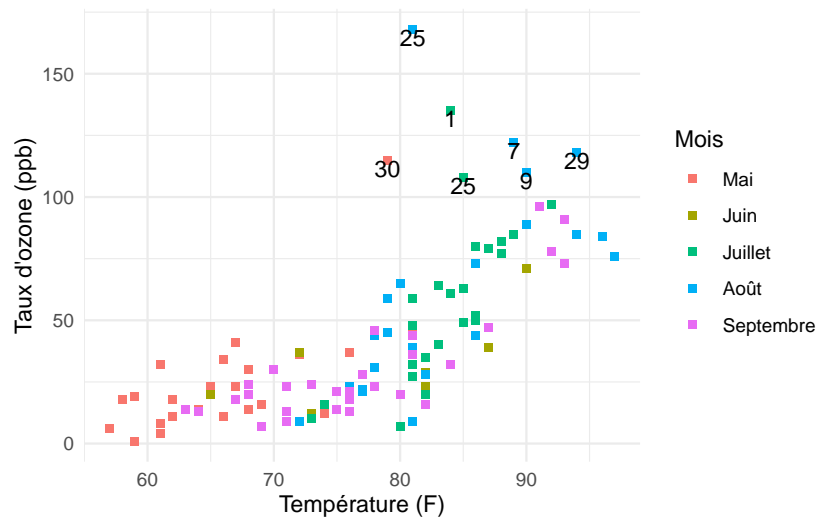
```
ggplot(dataairquality, aes(x=Temp, y=Ozone, colour=Mois)) + geom_point() +
  geom_smooth(method="lm", formula=y~x, se=FALSE) +
  xlab("Température (F)") + ylab("Taux d'ozone (ppb)") +
  theme_minimal() +
  scale_colour_discrete(labels=
    c("Mai", "Juin", "Juillet", "Août", "Septembre"))
```



### 4.3.2 Ajouter une annotation

Nous voulons savoir quels sont les jours dont le taux d'ozone est supérieur à 100.

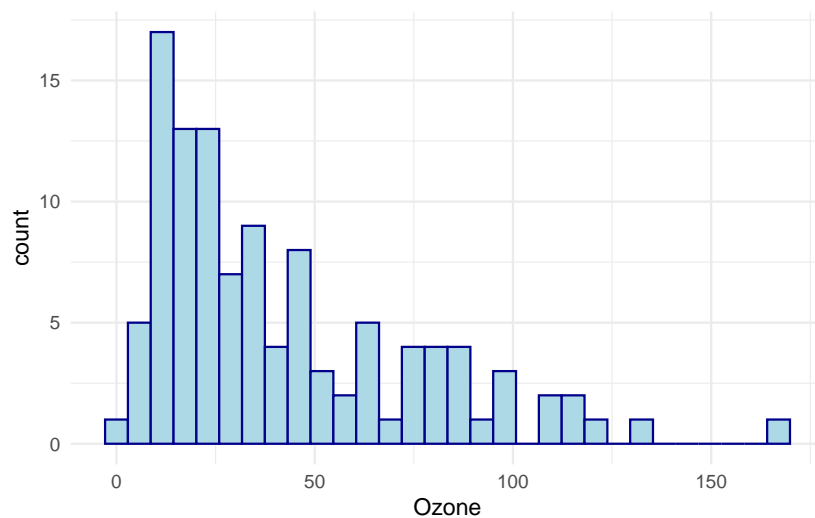
```
data_ozone_sup_100 <- dataairquality %>% filter(Ozone>100)
ggplot(dataairquality, aes(x=Temp, y=Ozone)) +
  geom_point(pch=15, aes(colour=Mois)) +
  geom_text(aes(label=Day, vjust="right"), data=data_ozone_sup_100) +
  xlab("Température (F)") + ylab("Taux d'ozone (ppb)") + theme_minimal() +
  scale_colour_discrete(labels =
    c("Mai", "Juin", "Juillet", "Août", "Septembre"))
```



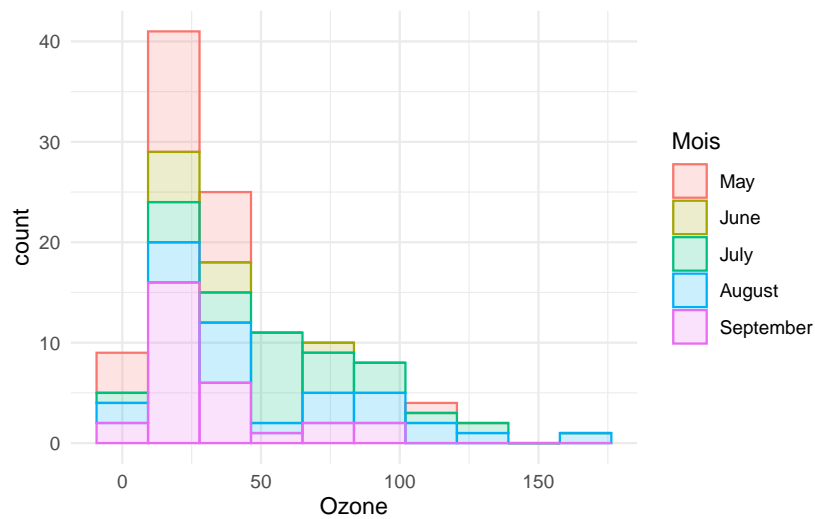
## 4.4 Construction d'un histogramme et d'une courbe de densité

```
ggplot(dataairquality, aes(x=Ozone)) +
  geom_histogram(color="darkblue", fill="lightblue") +
  theme_minimal()
```

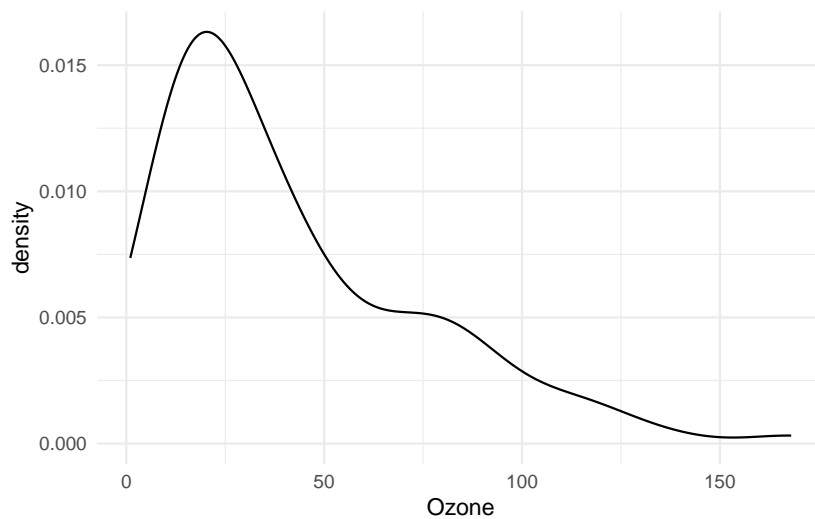
`stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



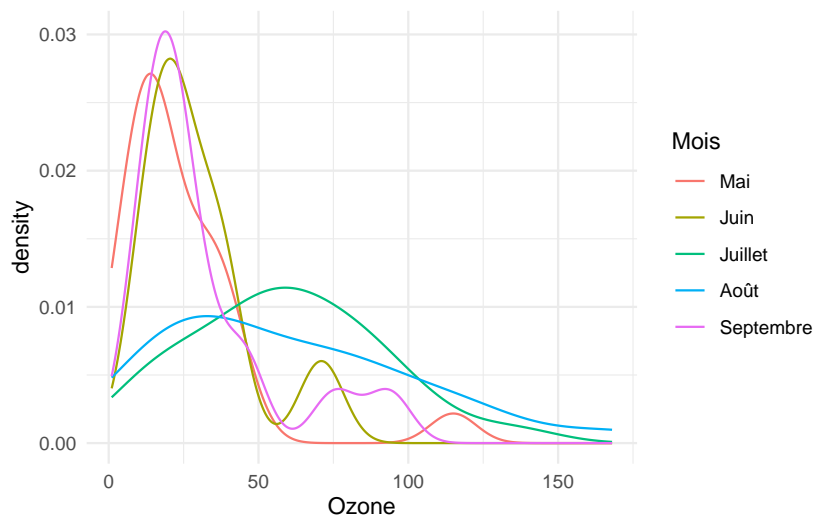
```
ggplot(dataairquality, aes(x=Ozone)) +
  geom_histogram(alpha=0.2, bins=10, aes(colour=Mois, fill=Mois)) +
  theme_minimal()
```



```
ggplot(dataairquality, aes(x=Ozone)) +  
  geom_density() + theme_minimal()
```



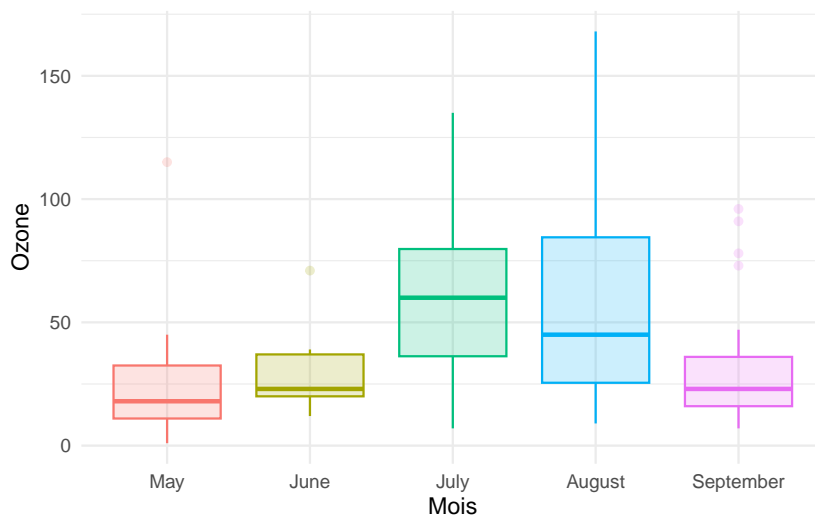
```
ggplot(dataairquality, aes(x=Ozone)) +  
  stat_density(aes(colour=Mois), geom="line", position="identity") +  
  theme_minimal() +  
  scale_colour_discrete(labels =  
    c("Mai", "Juin", "Juillet", "Août", "Septembre"))
```



## 4.5 Construction d'une boîte à moustaches

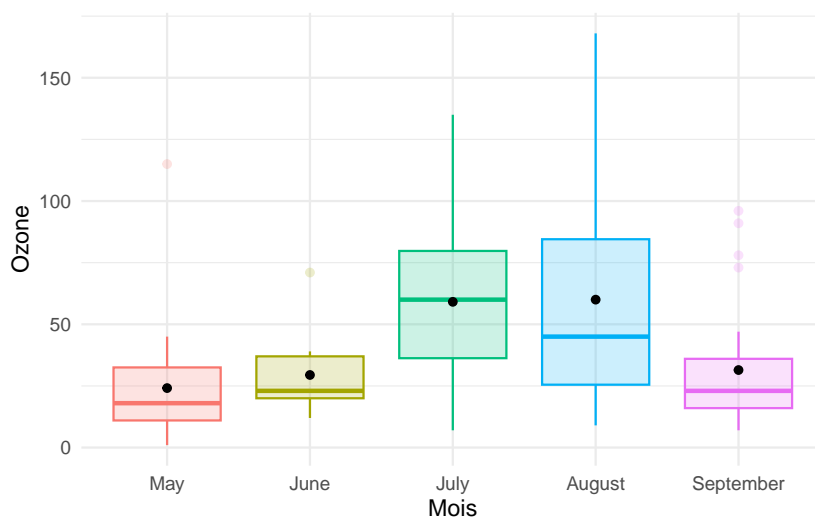
Traçons les boîtes à moustaches de la variable Ozone pour chaque mois avec la fonction `geom_boxplot()`.

```
ggplot(dataairquality, aes(x=Mois, y=Ozone)) +
  geom_boxplot(alpha=0.2, aes(colour=Mois, fill=Mois),
    show.legend = FALSE) +
  theme_minimal()
```



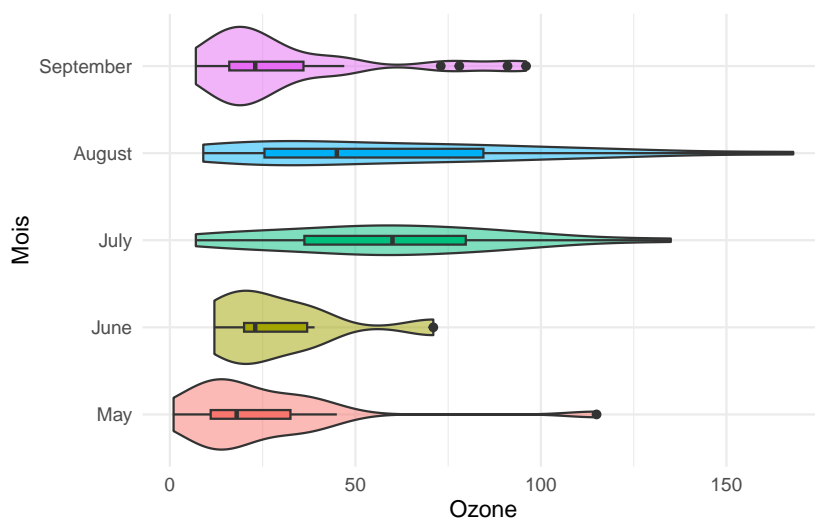
Ajoutons un point correspondant au taux d'ozone moyen pour chaque mois.

```
ggplot(dataairquality, aes(x=Mois, y=Ozone)) +
  geom_boxplot(alpha=0.2, aes(colour=Mois, fill=Mois),
    show.legend = FALSE) +
  stat_summary(fun=mean, geom="point") +
  theme_minimal()
```



Traçons des violin plots avec la fonction `geom_violin()`.

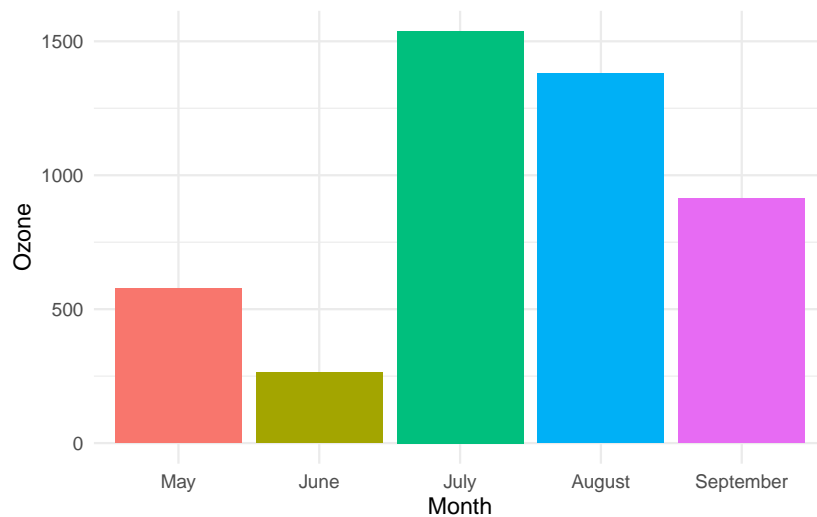
```
ggplot(dataairquality, aes(x=Ozone, y=Mois)) +
  geom_violin(alpha=0.5, aes(fill=Mois), show.legend = FALSE) +
  geom_boxplot(width=0.1, aes(fill=Mois), show.legend = FALSE) +
  theme_minimal()
```



## 4.6 Construction d'un diagramme en barres

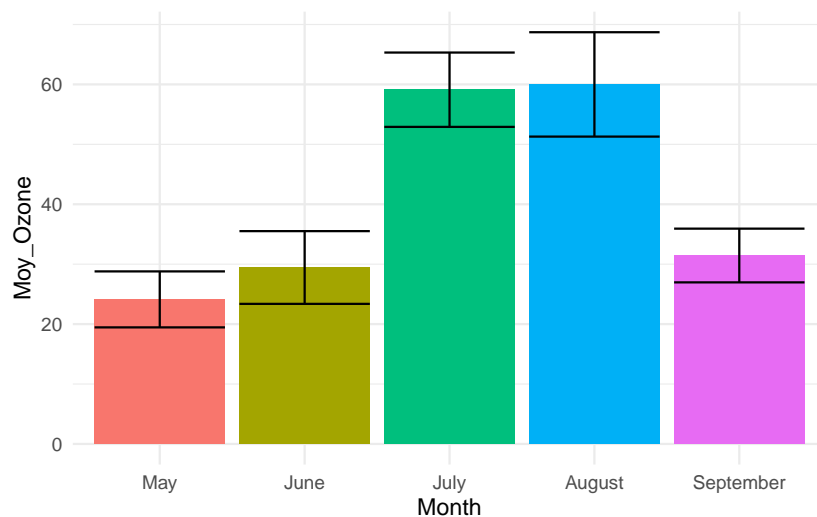
On peut utiliser `geom_bar()` pour effectuer un diagramme en barre.

```
ggplot(dataairquality, aes(y=Ozone, x=Mois)) +
  geom_bar(stat="identity", aes(fill=Mois), show.legend=FALSE) +
  xlab("Month") +
  theme_minimal()
```



On peut ajouter les barres d'erreur au diagramme.

```
dataresumme <- dataairquality %>% group_by(Mois) %>%
  summarise(n=n(), Moy_Ozone = mean(Ozone, na.rm=TRUE),
            SE_Ozone=sd(Ozone, na.rm=TRUE)/sqrt(n))
ggplot(dataresumme, aes(y=Moy_Ozone, x=Mois)) +
  geom_bar(stat="identity", aes(fill=Mois), show.legend=FALSE) +
  xlab("Month") +
  geom_errorbar(aes(ymin=Moy_Ozone-SE_Ozone, ymax=Moy_Ozone+SE_Ozone)) +
  theme_minimal()
```



Vous pouvez installer l'extension `ggstatsplot` (<https://indrajeetpatil.github.io/ggstatsplot/>) permettant de créer des graphiques enrichis par des informations statistiques (résultats de tests).

## 4.7 Les addins `ggplotAssist` et `esquisse`

Si vous souhaitez apprendre à faire des graphiques avec `ggplot2`, vous pouvez utiliser l'addin `ggplotAssist`. Vous devez installer la librairie `ggplotAssist` du CRAN.

```
library(ggplotAssist)
```

Un autre addin intéressant est l'addin *esquisse*. C'est un package permettant de visualiser ses données en générant des graphiques avec le package *ggplot2* ainsi que le code associé (application Shiny).

```
library(esquisse)
```

# Chapitre 5

## Un petit détour par le package broom

### 5.1 Pourquoi utiliser le package broom ?

- Utilisé pour les sorties de modèles statistiques
- Résume l'information dans un objet tibble
- Facilite la représentation graphique

```
library(broom)
```

Prenons un exemple. On souhaite modéliser le taux d'ozone en fonction de la température. Effectuons un modèle de régression linéaire.

```
exreg <- lm(Ozone~Temp, data=dataairquality)
```

### 5.2 Résumer la sortie d'un modèle statistique : La fonction tidy()

Prenons la sortie du modèle de régression linéaire. Résumons la sortie :

```
tidy(exreg, conf.int = TRUE)
```

```
# A tibble: 2 x 7
```

term	estimate	std.error	statistic	p.value	conf.low	conf.high
<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1 (Intercept)	-148.	18.8	-7.87	2.76e-12	-185.	-110.
2 Temp	2.44	0.239	10.2	1.55e-17	1.96	2.91

### 5.3 Récapitulatif d'un modèle : La fonction glance()

```
glance(exreg)
```

```
# A tibble: 1 x 12
```

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1 0.488	0.483	23.9	104.	1.55e-17	1	-509.	1024.	1032.



```
# i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

## 5.4 Ajouter des informations sur le tableau de données : La fonction augment()

```
augment(exreg)

# A tibble: 111 x 8
  Ozone Temp .fitted .resid .hat .sigma .cooksd .std.resid
  <int> <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1    41    67  15.8  25.2  0.0207  23.9  0.0120    1.07
2    36    72  28.0   8.03  0.0124  24.0  0.000714    0.338
3    12    74  32.8 -20.8  0.0104  23.9  0.00405   -0.876
4    18    62   3.58  14.4  0.0340  24.0  0.00662    0.613
5    23    65  10.9  12.1  0.0254  24.0  0.00342    0.513
6    19    59  -3.74  22.7  0.0444  23.9  0.0219    0.972
7     8    61   1.14   6.86  0.0372  24.0  0.00165    0.292
8    16    69  20.7  -4.65  0.0167  24.0  0.000328   -0.196
9    11    66  13.3  -2.34  0.0229  24.0  0.000114   -0.0988
10   14    68  18.2  -4.21  0.0186  24.0  0.000300   -0.178
# i 101 more rows
```

## 5.5 Pour aller plus loin

Effectuons un modèle de régression linéaire pour chaque mois.

```
dataairquality %>% group_by(Mois) %>%
  group_map(~tidy(lm(. $Ozone~. $Temp, data=.x)), conf.int=TRUE)

[[1]]
# A tibble: 2 x 5
  term          estimate std.error statistic p.value
  <chr>          <dbl>    <dbl>    <dbl>    <dbl>
1 (Intercept)  -116.      38.8      -3.00  0.00666
2 .$Temp         2.11     0.581       3.63  0.00146

[[2]]
# A tibble: 2 x 5
  term          estimate std.error statistic p.value
  <chr>          <dbl>    <dbl>    <dbl>    <dbl>
1 (Intercept)  -92.0     51.3      -1.79  0.116
2 .$Temp         1.55     0.653       2.38  0.0491

[[3]]
# A tibble: 2 x 5
  term          estimate std.error statistic p.value
  <chr>          <dbl>    <dbl>    <dbl>    <dbl>
1 (Intercept) -373.      84.5      -4.42  0.000184
```

```
2 .$Temp          5.15      1.01      5.12 0.0000305
```

```
[[4]]
```

```
# A tibble: 2 x 5
```

	term	estimate	std.error	statistic	p.value
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>
1	(Intercept)	-240.	86.4	-2.78	0.0112
2	.\$Temp	3.58	1.03	3.49	0.00220

```
[[5]]
```

```
# A tibble: 2 x 5
```

	term	estimate	std.error	statistic	p.value
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>
1	(Intercept)	-149.	23.7	-6.30	0.000000951
2	.\$Temp	2.35	0.306	7.68	0.0000000295

# Bibliographie

- [1] J.M. CHAMBERS et al. "Graphical Methods for Data Analysis". In : *The Wadsworth Statistics/Probability Series*. Boston, MA: Duxury (1983).
- [2] W. CHANG. *R Graphics Cookbook : Practical Recipes for Visualizing Data*. O'Reilly Media, 2012. ISBN : 9781449363109.
- [3] P. TEETOR. *R Cookbook : Proven Recipes for Data Analysis, Statistics, and Graphics*. O'Reilly Media, 2011. ISBN : 9781449307264.
- [4] H. WICKHAM. *Advanced R*. CRC Press, 2014.
- [5] H. WICKHAM et G. GROLEMUND. *R for Data Science*. O'Reilly Media, 2017. ISBN : 9781491910399.
- [6] Hadley WICKHAM. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016. ISBN : 978-3-319-24277-4. URL : <https://ggplot2.tidyverse.org>.
- [7] Hadley WICKHAM et Jennifer BRYAN. *readxl: Read Excel Files*. R package version 1.3.1. 2019. URL : <https://CRAN.R-project.org/package=readxl>.
- [8] Hadley WICKHAM et al. "Welcome to the tidyverse". In : *Journal of Open Source Software* 4.43 (2019), p. 1686. DOI : [10.21105/joss.01686](https://doi.org/10.21105/joss.01686).