

**-SUP-
en poche**

ÉCO

L1 / L2

Statistiques descriptives et probabilités

**Licence, BUT, préparation aux
concours et formation continue**

Yves Croissant

- ☒ Fiches et résumés de cours
- ☒ Exercices avec corrigés
- ☒ Plus de 100 questions
- ☒ Applications sur le logiciel R

**-SUP-
en poche**

ÉCO

L1 / L2

Statistiques descriptives et probabilités

Yves Croissant

Pour toute information sur notre fonds et les nouveautés dans votre domaine de spécialisation, consultez notre site web : www.deboecksuperieur.com

© De Boeck Supérieur s.a., 2021
Rue du Bosquet, 7 - B-1348 Louvain-la-Neuve

Tous droits réservés pour tous pays.

Il est interdit, sauf accord préalable et écrit de l'éditeur, de reproduire (notamment par photocopie) partiellement ou totalement le présent ouvrage, de le stocker dans une banque de données ou de le communiquer au public, sous quelque forme et de quelque manière que ce soit.

Dépôt légal :
Bibliothèque Nationale, Paris : avril 2021
Bibliothèque royale de Belgique, Bruxelles : 2021/13647/065

ISSN : 2566-2708
ISBN : 978-2-8073-2756-6

Sommaire

Introduction	1
Partie 1 Introduction à R	
Installation	5
1 Visualiser des données avec <code>ggplot2</code>	11
2 Transformer les données avec <code>dplyr</code>	26
Partie 2 Distribution à un caractère	
3 Tableaux et graphiques pour une distribution à un caractère ..	43
4 Caractéristiques des distributions à un caractère	71
Partie 3 Distributions à deux caractères	
5 Représentation des séries à deux caractères	99
6 Liaisons entre deux caractères	115
Partie 4 Séries chronologiques	
7 Indices	133
8 Analyse des séries chronologiques	147
Partie 5 Probabilités et variables aléatoires	
9 Introduction à la théorie des probabilités	171
10 Variable aléatoire	187

Partie 6 Fonctions de distribution de probabilités

11	Variables aléatoires discrètes	205
12	Variables aléatoires continues	219
	Index	236

Introduction

Ce livre s'adresse aux étudiants de L1 et de L2 Économie et AES. Il correspond aux deux premiers cours de statistiques traditionnellement dispensés en licence, statistiques descriptives et probabilités. Il utilise plusieurs jeux de données, souvent relativement volumineux (plusieurs centaines de milliers de lignes pour certains d'entre eux). Nous pensons qu'il est souhaitable que l'étudiant, dès le début de son cursus, se confronte avec des jeux de données réels et ne se contente pas d'utiliser des petits jeux de données à l'intérêt purement pédagogique.

Ce livre utilise **R** pour les applications. **R** est un logiciel libre que vous pouvez installer gratuitement à partir d'Internet.

Vous trouverez dans ce livre de nombreuses lignes de code que vous pouvez utiliser dans **R** afin de reproduire certains tableaux et graphiques. Plus précisément, ce livre utilise ce que l'on peut qualifier d'approche moderne de **R**, développée par Hadley Wickham. Les fonctions qu'il a développées sont connues sous le nom de *tidyverse*. Elles sont plus cohérentes, plus intuitives et souvent plus rapides que les fonctions de **R** qu'elles remplacent. Le livre d'Hadley Wickham et de Garrett Golemund (Wickham et Golemund, 2018) *R for data science*¹ est disponible en ligne (en anglais) et constitue un complément indispensable à ce livre pour les étudiants qui souhaiteraient approfondir leurs compétences en **R**.

L'approche d'apprentissage de **R** choisie dans ce livre est progressive. La première partie du livre est consacrée à une introduction à **R**, axée sur la réalisation de graphiques et sur la manipulation de tableaux de données. Dans les chapitres suivants, à chaque fois qu'une nouvelle fonction est utilisée, elle est décrite sommairement dans un encadré. Un index permet de repérer les fonctions décrites et de retrouver aisément l'encadré explicatif correspondant.

Il existe de très nombreux manuels traitant des statistiques descriptives et des probabilités. Nous avons consulté en particulier, au cours de la rédaction de cet ouvrage, Wonnacott et Wonnacott (1998), Anderson *et al.* (2013), Py (2007), Goldfarb et Pardoux (2011) et Compte et Gaden (2000).

1. <https://r4ds.had.co.nz/>

Partie 1

Introduction à R

SOMMAIRE

Installation

1 Visualiser des données avec ggplot2

2 Transformer les données avec dplyr

Installation

L'utilisation de **R** nécessite l'installation de deux logiciels :

- ◆ **R** en tant que tel, qui est librement disponible au téléchargement à l'adresse (<https://cran.r-project.org/>),
- ◆ **Rstudio**, la version « Desktop », gratuite est disponible au téléchargement à l'adresse (<https://rstudio.com/products/rstudio/download/>)

Ces deux logiciels sont libres et multi-plateformes, c'est-à-dire que des versions pour Windows, Mac et Linux sont disponibles.

Il convient de bien distinguer les deux; **R** est le logiciel en tant que tel, **Rstudio** est ce qu'on appelle un environnement de développement intégré, c'est-à-dire une interface permettant de travailler efficacement avec **R**. Vous n'utiliserez jamais **R** directement, mais via **Rstudio**. Lorsque vous ouvrez **Rstudio**, vous obtenez l'écran représenté sur la figure 1.

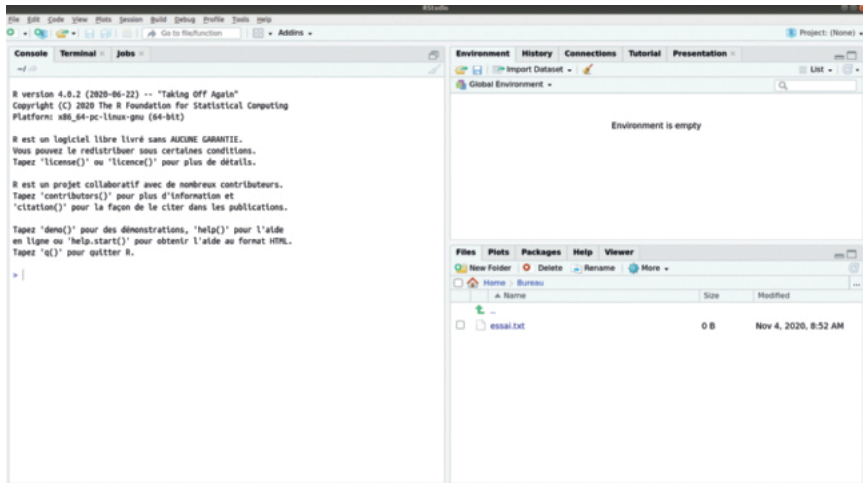


Figure 1 Fenêtre de Rstudio

L'écran est divisé en 3 :

- ◆ à gauche, la console, c'est l'endroit où vous pourrez interagir avec **R**, en écrivant une expression après l'invite de commande (le signe `>`); essayez par exemple d'écrire `1 + 1` et validez votre expression avec la touche `Enter`, vous devez alors obtenir `[1] 2`. D'autres onglets sont

accessibles, Terminal et Jobs mais vous n'en aurez pas l'usage dans un premier temps,

- ◆ en haut à droite, vous avez accès à l'environnement, c'est-à-dire à votre espace de travail. Pour l'instant, l'environnement est vide, mais au fur et à mesure que vous travaillerez avec **R**, vous allez créer des **objets** (nous définirons ce terme ultérieurement) qui apparaîtront alors dans cet environnement. D'autres onglets sont disponibles, mais comme précédemment, nous ne les utiliserons pas,
- ◆ en bas à droite, plusieurs onglets très utiles sont disponibles :
 - Files présente l'arborescence de votre **répertoire de travail**; dans la copie d'écran correspondant à la figure 1, ce répertoire est le bureau, qui ne contient qu'un fichier, appelé `essai.txt`,
 - Plots est la fenêtre dans laquelle les graphiques que nous allons créer apparaîtront,
 - Packages est l'onglet qui va nous permettre d'installer et de charger en mémoire des **packages**, c'est-à-dire des librairies qui étendent les capacités de **R**,
 - Help est l'onglet dans lequel apparaîtront les pages d'aides.

En plus de ces onglets, vous disposez d'un menu déroulant, ainsi que d'icônes qui permettent en particulier d'ouvrir et de sauvegarder un fichier. Pour disposer de l'ensemble des ressources utilisées dans ce livre, vous devez :

- ◆ installer les trois packages **remotes**, **tidyverse** et **descstat**; pour cela, sélectionnez l'onglet Packages dans la fenêtre en bas à droite, puis l'onglet Install; apparaît alors une fenêtre dans laquelle vous pouvez taper le nom du ou des packages que vous souhaitez installer, c'est-à-dire ici `remotes tidyverse descstat`. Cliquez alors sur Install de manière à installer les trois packages. Cette opération nécessite bien entendu que vous soyez connecté à Internet,
- ◆ afin d'installer le package **statecoaes**, taper dans la console (la fenêtre de gauche)¹ :

```
remotes::install_github("ycroissant/statecoaes")
```

tidyverse est un méta package qui contient plusieurs packages développés par Hadley Wickham, en particulier :

- ◆ **ggplot2** qui permet de réaliser des graphiques et qui sera décrit dans le chapitre 1,

1. Si le texte dans la console demande une mise à jour, cliquez sur 1, puis Enter, pour télécharger tous les packages à mettre à jour.

- ◆ **dplyr** qui permet de réaliser différentes opérations sur les tableaux de données et dont les principales fonctionnalités seront présentées dans le chapitre 2.

descstat est un package écrit par l'auteur de ce livre qui fournit des fonctions permettant de créer facilement des tableaux statistiques, de les représenter graphiquement et de calculer à partir de ceux-ci différentes statistiques.

Ces deux packages sont stockés sur le serveur **cran** (pour Comprehensive R Archive Network), qui est le lieu où sont stockées les extensions de **R** et ils peuvent donc être aisément installés avec l'interface de **Rstudio**.

remotes est utilisé uniquement pour pouvoir installer le package **statecoaes**. Ce dernier contient les données utilisées dans ce livre et ces données sont trop volumineuses pour que ce package puisse être stocké sur **cran**. C'est la raison pour laquelle on doit utiliser une commande spéciale (`install_github`) pour l'installer.

L'installation des packages n'est faite qu'une seule fois. Il ne reste alors plus qu'à créer un nouveau script, c'est-à-dire un fichier qui contiendra les instructions de **R**. À cet effet, vous pouvez indifféremment :

- ◆ utiliser l'entrée de menu `File -> New File -> R Script`,
- ◆ cliquer sur l'icône juste en dessous de l'entrée `File` du menu,
- ◆ utiliser le raccourci clavier `Ctrl + Maj + N` (ce qui signifie appuyer simultanément sur les trois touches).

La fenêtre de gauche est alors divisée horizontalement en deux, le haut étant occupé par un fichier (pour l'instant vide) appelé `Untitled1`. Sauvegardez ce fichier, par exemple en cliquant sur l'icône représentant une disquette et indiquez comme nom `Deaton`, sans extension, l'extension `.R` qui permet de reconnaître les fichiers de script **R** étant automatiquement ajoutée. Écrivez ensuite dans ce fichier :

```
library("tidyverse")
library("descstat")
library("statecoaes")
```

library



La commande `library` permet de charger en mémoire une extension déjà installée sur votre ordinateur ; vous n'installez qu'une seule fois l'extension, mais vous devez la charger en mémoire au début de chaque session. L'argument principal de `library` (et le seul que vous utiliserez la plupart du temps) est le nom de la librairie, exprimé sous la forme d'une chaîne de caractères, c'est-à-dire entre guillemets.

Vous prendrez l'habitude de charger systématiquement ces trois librairies chaque fois que vous voudrez reproduire des exemples du livre. Pour évaluer ces expressions, positionnez le curseur sur la première ligne (pas nécessairement au début de celle-ci) et saisissez la combinaison de touche `Ctrl + Enter`. Celle-ci a deux effets :

- ◆ elle évalue la ligne concernée (ici `library("tidyverse")`),
- ◆ elle positionne le curseur sur la ligne suivante (`library("statecoaes")`).

Vous pouvez ainsi aisément évaluer successivement un ensemble de ligne simplement en laissant la touche `Ctrl` enfoncée et en appuyant de manière répétée sur la touche `Enter`.

Écrivez ensuite dans votre script et évaluez :

Deaton

```
## # A tibble: 179 x 8
##   code  nom    region incgp   gdp    pop    le subnom
##   <chr> <chr>  <fct>  <fct>  <dbl>  <dbl> <dbl> <chr>
## 1 AFG   Afgha~ South ~ Low ~   2242. 3.72e7  64.5 <NA>
## 2 AGO   Angola Sub-Sa~ Lowe~   7097. 3.08e7  60.8 <NA>
## 3 ALB   Alban~ Europe~ Uppe~  13834. 2.87e6  78.5 <NA>
## 4 ARE   Unite~ Middle~ High~  68549. 9.63e6  77.8 <NA>
## 5 ARG   Argen~ Latin ~ Uppe~  23283. 4.45e7  76.5 <NA>
## # ... with 174 more rows
```

Deaton est un tibble. Il s'agit d'un tableau à deux dimensions, chaque ligne correspondant à une observation et chaque colonne à une variable.

tibble

Un **tibble** est une version améliorée d'un `data.frame` (littéralement « cadre de données » en français), qui est la structure utilisée par **R** pour stocker des données. En particulier, l'impression des tibbles est particulièrement agréable dans la mesure où elle est limitée à un nombre de lignes et de colonnes qui permettent l'affichage dans la fenêtre. L'affichage indique :



- ◆ la dimension du tableau : 179 x 8 ce qui signifie 179 lignes (observations) et 8 colonnes (variables),
- ◆ le nom des variables : `code`, `nom`, etc.,
- ◆ le type des variables :
 - `chr` signifie chaîne de caractères (character en anglais),
 - `fct` signifie variable catégorielle (factor en anglais),
 - `dbl` signifie nombre réel (double en anglais).

Lorsque l'on saisit directement le nom du **tibble** (ici `Deaton`), c'est comme si on saisisait `print(Deaton)`. `print` est une fonction qui peut prendre comme arguments, en plus du **tibble**, `n` pour définir le nombre de lignes à imprimer et `width` pour définir la largeur de la portion du tableau qui sera imprimée.

La commande :

```
print(Deaton, n = 2, width = 40)
```

```
## # A tibble: 179 x 8
##   code  nom    region incgp  gdp    pop
##   <chr> <chr>  <fct>  <fct> <dbl>  <dbl>
## 1 AFG   Afgha~ South~ Low   ~ 2242.  3.72e7
## 2 AGO   Angola Sub-S~ Lowe~ 7097.  3.08e7
## # ... with 177 more rows, and 2 more
## #   variables: le <dbl>, subnom <chr>
```

permet d'imprimer les deux premières lignes du tableau et les 6 premières variables. Les deux variables supplémentaires non imprimées (`le` et `subnom`) sont uniquement listées, avec indication de leur type.

Dans ce chapitre, nous utiliserons les variables suivantes :

- ◆ `gdp` : la richesse (mesurée par le produit intérieur brut) par habitant,
- ◆ `le` : l'espérance de vie,
- ◆ `pop` : la population,
- ◆ `subnom` : un sous-ensemble de noms de pays,

- ◆ `region` : les grandes régions du monde,
- ◆ `incgp` : la classification des pays selon le niveau de revenu.

Angus Deaton, prix Nobel d'économie 2015, a publié en 2016 un ouvrage intitulé « La grande évasion » (Deaton, 2016), dans lequel il décrit l'évasion de la pauvreté d'une grande partie de la population mondiale au cours du XX^{ème} siècle. Il illustre son propos par plusieurs graphiques, dont celui reproduit sur la figure 1.1, qui présente la relation entre la richesse par tête et l'espérance de vie.

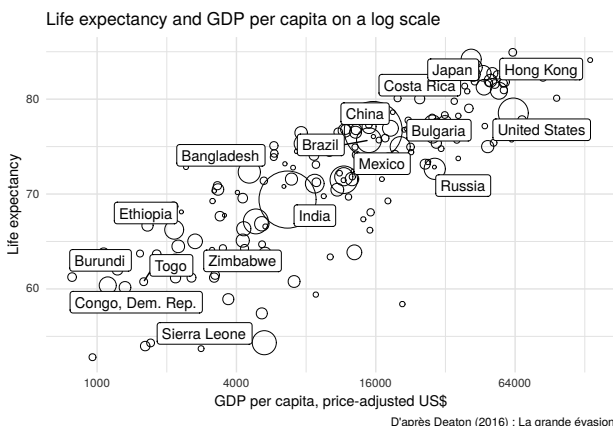


Figure 1.1 Espérance de vie et richesse

Analysons en détail ce graphique :

- ◆ l'axe des abscisses indique la richesse (mesurée par le produit intérieur brut) par habitant,
- ◆ l'axe des ordonnées montre l'espérance de vie,
- ◆ les éléments graphiques représentés sont des points et chaque point correspond à un pays,
- ◆ la taille du point est proportionnelle à la population de chaque pays,
- ◆ le nom de certains pays est noté sur le graphique,
- ◆ l'échelle horizontale est logarithmique.

1 Introduction

Nous allons utiliser la librairie **ggplot2** dont le nom signifie **grammar of graphics**. Cette “grammaire” est constituée de différents éléments :

- ◆ **data** : le tableau qui contient les données que l'on souhaite représenter graphiquement,
- ◆ **mapping** : l'association entre des éléments visuels du graphique (des coordonnées, des couleurs par exemple) et des variables du tableau,
- ◆ **geom** : les éléments graphiques, par exemple des points, des lignes, des barres, du texte, etc.
- ◆ **scale** : les échelles, qui peuvent être des échelles concernant les axes (limites, graduations, etc.), mais également des échelles de couleur, de taille, etc.
- ◆ **stat** : des fonctions qui transforment les données brutes du tableau avant de les tracer sur le graphique,
- ◆ **facets** : la création de sous-graphiques en fonction des modalités d'une ou de plusieurs variables catégorielles,
- ◆ **coord** : le système de coordonnées, par exemple cartésiennes, polaires, ou encore projetées dans le cas de cartes,
- ◆ **position** : la position des éléments par rapport aux coordonnées.

Le package **ggplot2** fournit la commande **ggplot** qui est la fonction principale permettant de créer un graphique. On indique comme argument le tableau **Deaton**. En évaluant :

```
ggplot(Deaton)
```

ggplot

Tout graphique créé avec **ggplot2** doit commencer par un appel à la commande **ggplot**. Celle-ci prendra en général deux arguments :

- ◆ le premier, appelé **data**, indique les données que l'on veut représenter,
- ◆ le second, appelé **mapping** va permettre de spécifier de manière globale l'association entre des variables du tableau et des caractéristiques visuelles du graphique.



on obtient dans l'onglet Plots de la fenêtre inférieure droite un fond graphique entièrement vide. Renseignons désormais l'argument `mapping`, en associant, à l'aide de la fonction `aes`, `gdp` et `le` aux axes des abscisses et des ordonnées.

aes

La fonction `aes` permet d'associer des variables du tableau à des caractéristiques du graphique. À cette fin, on utilise une syntaxe du type `car = var` où `car` est la caractéristique graphique et `var` la variable du tableau que l'on souhaite associer à cette caractéristique. Chaque caractéristique est appelée un *aesthetic*, dont les principaux sont :



- ◆ `x`, `y` : l'axe des abscisses et des ordonnées,
- ◆ `color`, `fill` : la couleur (pour une ligne, un point par exemple) et la couleur de remplissage (pour une barre par exemple),
- ◆ `linetype` : le type de ligne (trait plein, pointillé,...),
- ◆ `shape` : la forme d'un point,
- ◆ `size` : la taille,
- ◆ `label` : un texte.

Comme on souhaite représenter la relation entre richesse et espérance de vie, on indique ici en abscisses `gdp` et en ordonnées `le`.

```
ggplot(Deaton, aes(x = gdp, y = le))
```

Notons que l'on aurait obtenu le même résultat en utilisant `aes(gdp, le)` parce que `x` et `y` sont les deux premiers arguments de la fonction `aes`. Désormais, les axes figurent sur le graphique et les valeurs indiquées sont adaptées à l'étendue de nos données.

Pour représenter la relation entre richesse et espérance de vie par des points, il faut ajouter un `geom`, ici `geom_point`. Il s'agit d'une fonction, il faut donc mettre des parenthèses, même si elles sont ici vides. `geom_point` est un *layer* (une couche en français). On va ainsi pouvoir construire un graphique pas à pas en ajoutant des couches successives. Avec **ggplot2**, les différentes couches seront superposées à l'aide du signe `+`. Le résultat est présenté sur la figure 1.2.

```
ggplot(Deaton, aes(x = gdp, y = le)) + geom_point()
```

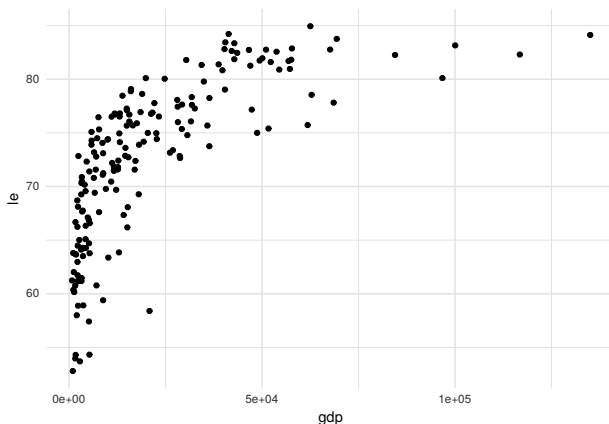


Figure 1.2 Deaton (1), nuage de points de l'espérance de vie en fonction de la richesse par tête

geom_point



`geom_point` permet de tracer des points. La position des points est donnée par les *aesthetics* `x` et `y` ; on peut également définir d'autres caractéristiques comme la taille (`size`), la couleur (`color`) ou la forme (`shape`), cette dernière caractéristique étant définie par un nombre entier. Comme beaucoup de `geom`, `geom_point` a pour argument principal `mapping` et `data`. Le premier permet d'associer *localement*, c'est-à-dire juste pour cette fonction, des variables à des caractéristiques du graphique, en utilisant la fonction `aes`. Le second permet de définir une table de données spécifique pour cette fonction.

2 Conseils sur le code

Lorsqu'on ajoute successivement un grand nombre de couches, les lignes de codes peuvent rapidement devenir très longues. Il est conseillé de les couper, en s'assurant que la césure définit une instruction incomplète et donc non évaluée. Par exemple, la césure suivante est correcte :

```
ggplot(Deaton, aes(x = gdp, y = le)) +  
  geom_point()
```

alors que la suivante ne l'est pas, car la première ligne constitue une instruction complète qui est évaluée et la seconde, qui commence par un +, est incorrecte et renvoie une erreur.

```
ggplot(Deaton, aes(x = gdp, y = le))
+ geom_point()
```

Il est conseillé de commenter le code. Pour **R**, tout ce qui figure après le signe # est considéré comme un commentaire, ce qui permet d'écrire des lignes entières de commentaires, ou des commentaires sur la même ligne que le code :

```
# Un premier graphique avec ggplot
ggplot(Deaton, aes(x = gdp, y = le)) +
  geom_point() # geom_point permet d'ajouter des points
```

Il est très courant de faire des erreurs syntaxiques, une des plus courantes consistant à oublier de fermer une parenthèse ouvrante :

```
ggplot(Deaton, aes(x = gdp, y = le) + geom_point())
```

Rstudio aide à éviter ces erreurs :

- ◆ en écrivant automatiquement une parenthèse fermante à chaque fois qu'une parenthèse ouvrante est entrée,
 - ◆ en signalant les erreurs syntaxiques par une croix rouge dans la marge.
- Très souvent, des erreurs de ce type se manifestent lorsque le code est évalué par le fait qu'un + apparaît dans la console plutôt que le signe > habituel. Dans ce cas, utilisez la touche d'échappement (Esc en haut à gauche du clavier), corrigez l'erreur et réévaluez le code.

3 Personnalisation du graphique

Dans le graphique de Deaton, la taille des points est proportionnelle à la population. Ce résultat peut être obtenu en associant la variable `pop` à la caractéristique graphique `size`. Dans le même temps, nous associons la forme des points à une constante (21) qui correspond à des points de forme ronde et non remplis.

```
ggplot(Deaton, aes(x = gdp, y = le, size = pop)) +  
  geom_point(shape = 21)
```

On peut également repérer les différentes régions du monde en associant la variable `region` à un attribut des points : cela peut être par exemple la forme (`shape`) ou la couleur (`colour`). Le résultat est présenté sur la figure 1.3.

```
ggplot(Deaton, aes(x = gdp, y = le, size = pop,  
  colour = region)) +  
  geom_point(shape = 21)
```

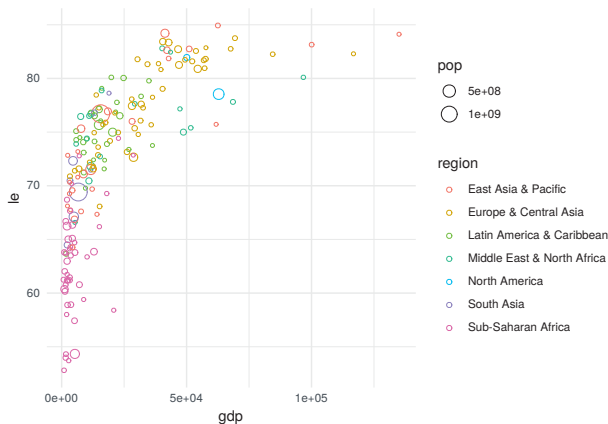


Figure 1.3 Deaton (2), points personnalisés dont la taille est liée à la population

4 Modification des échelles

scale_



Avec **ggplot2**, les échelles sont modifiées à l'aide d'instructions du type `scale_###` et souvent plus précisément `scale_###_discrete` ou `scale_###_continuous` pour personnaliser respectivement une échelle discrète et une échelle continue. Par exemple `scale_x_continuous` permet de définir les caractéristiques de l'axe des abscisses lorsque celui-ci est défini par une variable numérique continue. Parmi les nombreux arguments, notons `limits` qui permet de définir l'étendue de l'axe (par exemple `limits = c(0, 1000)` pour l'intervalle 0-1000) et `trans` qui permet de définir une transformation des valeurs représentées sur l'axe (par exemple `trans = "log10"` pour utiliser une transformation en logarithme décimal). Pour `scale_size_continuous`, on peut spécifier l'étendue des tailles (par exemple `range = c(1, 10)` pour obtenir des points de taille comprise entre 1 et 10) et `breaks` pour indiquer les valeurs qui seront représentées dans la légende.

On souhaite ici définir une échelle logarithmique pour l'axe des abscisses. On modifie également l'échelle de la taille des points, en indiquant l'étendue de la taille et les valeurs indiquées dans la légende (voir figure 1.4).

```
ggplot(Deaton, aes(x = gdp, y = le, size = pop,
                  colour = region)) +
  geom_point(shape = 21) +
  scale_x_continuous(trans = "log10") +
  scale_size_continuous(range = c(1, 10),
                        breaks = c(1E06, 1E07, 1E08,
                                   5E08, 1E09))
```

5 Utilisation de plusieurs geoms

Comme dans le graphique de Deaton, nous souhaitons écrire sur le graphique le nom de certains pays. Ceci est obtenu en utilisant un autre geom (`geom_text`) et en associant à l'*aesthetic* label la variable `subnom` (un sous-ensemble des noms de pays).

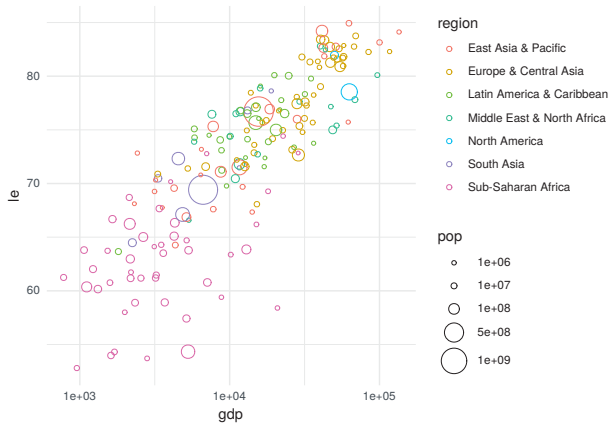


Figure 1.4 Deaton (3), modification des échelles

geom_text



`geom_text` écrit le texte figurant dans la variable associée à la caractéristique `label`, au point de coordonnées défini par les caractéristiques `x` et `y`. `geom_label` positionne le texte dans un rectangle.

```
ggplot(Deaton, aes(x = gdp, y = le, size = pop,
                    colour = region, label = subnom)) +
  geom_point(shape = 21) + geom_text() +
  scale_x_continuous(trans = "log10") +
  scale_size_continuous(range = c(1, 10),
                        breaks = c(1E6, 1E7, 1E8, 5E8, 1E9))
```

Nous constatons sur ce graphique que le nom du pays est écrit avec une police dont la taille est proportionnelle à la population et dont la couleur dépend de la région. Ceci est dû au fait que les associations de `pop` à `size` et de `region` à `colour` sont globales, elles s'appliquent donc à toutes les couches du graphique. Pour que les noms aient tous la même couleur et la même taille de police, on associe localement `pop` à `size` et `region` à `colour` dans `geom_point` afin que cette association ne soit effective que pour ce `geom` (voir figure 1.5).


```
dtn <- ggplot(Deaton, aes(x = gdp, y = le)) +
  geom_point(aes(size = pop, colour = region), shape = 21) +
  geom_text(aes(label = subnom), size = 2) +
  scale_x_continuous(trans = "log10") +
  scale_size_continuous(range = c(1, 10),
    breaks = c(1E6, 1E7, 1E8, 5E8, 1E9))
dtn
```

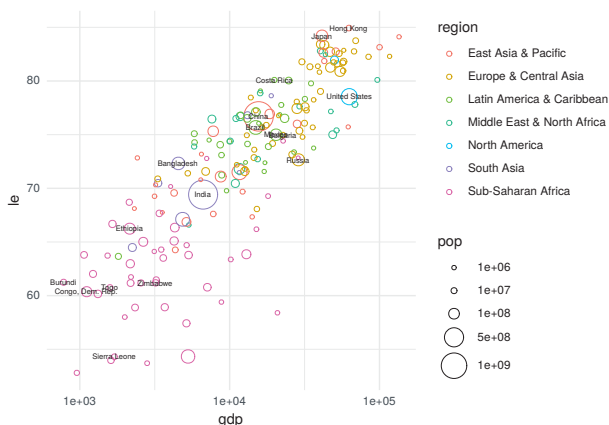


Figure 1.5 Deaton (4), plusieurs geom

Notez que, pour ce dernier graphique, on a introduit à gauche de la commande graphique `dtn <-`.

<-



<- est l'opérateur d'affectation (on peut aussi utiliser =). La valeur indiquée à droite de l'opérateur est affectée dans la variable indiquée à gauche : par exemple `a <- 2` signifie que l'on crée une nouvelle variable `a` à laquelle on affecte la valeur de 2. Si `a` existait au préalable (et valait par exemple 3), l'ancienne valeur est remplacée par la nouvelle. L'opérateur d'affectation peut être facilement obtenu à l'aide de la combinaison de touches `Alt + =` ou `Alt + _`.

Le graphique est alors stocké dans un objet appelé `dtn`. Dans ce cas rien ne s'affiche, mais l'objet `dtn` apparaît dans l'onglet `Environment` qui liste les objets disponibles dans votre espace de travail. Pour afficher le graphique, il

Statistiques descriptives et probabilités

Licence, BUT, préparation aux concours et formation continue

Cet ouvrage présente de façon synthétique les notions clés et les principaux outils de la statistique descriptive et des probabilités à partir de nombreuses applications sur le logiciel R. Sont ainsi traités : les distributions à un et à deux caractères(s), les indices (élémentaires et synthétiques) et les séries temporelles (tendance, CVS) et, pour les probabilités, le conditionnement et la dépendance, les variables aléatoires (discrètes, continues) et les lois de probabilité usuelles.

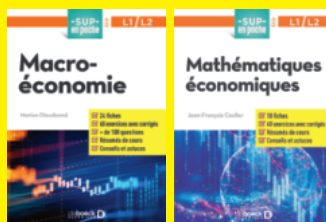
Chaque fiche contient :

- > **Les notions clés** afin de réviser les points essentiels.
- > **Des exemples et observations détaillés** pour illustrer les notions.
- > **Des points de méthode** qui résument la manière dont il convient d'utiliser les outils présentés dans la résolution d'un problème donné.
- > **Des exercices corrigés** élaborés à partir d'exemples d'applications concrètes.

Yves Croissant

est professeur de Sciences économiques à l'Université de La Réunion et membre de l'unité de recherche CEMOI.

DANS LA MÊME COLLECTION



ISSN : 2566-2708

ISBN : 978-2-8073-2756-6



Prix TTC : 18,90 €

deboeck B
SUPÉRIEUR

www.deboecksuperieur.com