Introduction aux Data Sciences

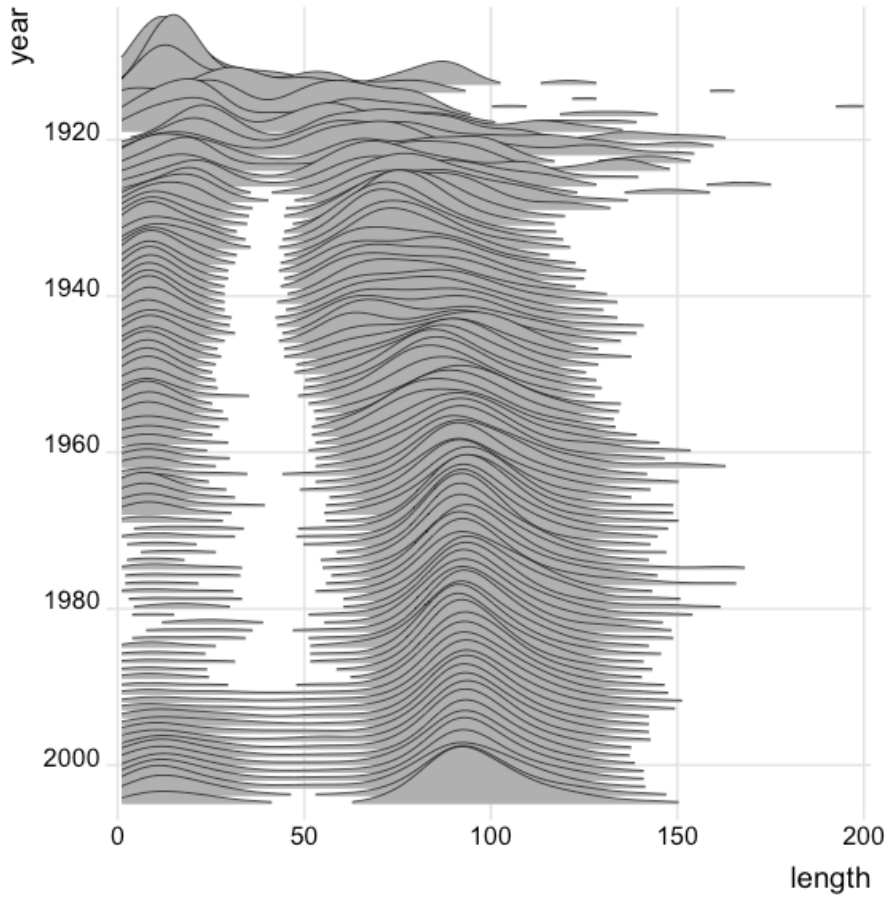
Christophe Benavent - Université Paris Dauphine

2022-10-16

Contents

Chapter 1

Avant propos



Ce bookdown présent les éléments d'un cours de data science avec r. Il est reproductible, on peut en cloner les élements à partir du repository. Le texte est encore hasardeux, les codes sont vérifiés. Il sera dynamique, modifié à mesure de nos cours, séminaires et ateliers.

L'illustration de couverture représente l'évolution de la longueur des films de la base Imbd et raconte en chiffres un aspect de l'histoire du cinéma. Jusqu'aux années 30, la longueur est hétérogène ensuite elle se se stabilise : les courts-métrages ont une durée de l'ordre de 15mn qui se raccourcit avec les décennies, ce genre menace de disparaître dans les années 80 et reprend du poil de la bête dans les années 2000. Les films longs voient leur longueur s'accroître et se stabiliser autour d'un peu moins de 100 mn, soit une heure et quarante minutes. On observera enfin qu'au cours des années 1990 les films de taille intermédiaires réapparaissent. On devinera dans cette évolution l'émergence de standards, ou de conventions. Les faits viennent au secours des théories…

Dans ce graphique il y a tous les éléments des data sciences contemporaines : un jeu de données riche et systématique, un modèle statistique fondamental avec la notion de densité de probabilité, une mesure, un critère de comparaison.

Les diagrammes ridges, c'est ainsi qu'on les appelent, sont inspirés de la pochette de l'album Unknown Pleasures de Joy division sorti en pleine période New Wave, en 1979. Un article de Vice en rappele l'origine et le destin du graphisme qu'on connait mieux imprimé sur des t-shirt que dans les cours de statistiques.

1.1 Plan du manuel

C'est un projet en cours, Les chapitres projetés sont les suivants. certains sont dans les limbes, d'autres ont pris consistances

- 1 - L'environnement r x
- 2 - Installation et prise en main x
- 3 - Usage de ggplot - uni et bivarié x
- 4 - Usage de ggplot - multivarié x
- 5 - Tables avec flex
- 6 - Modèles factoriels (Psych) x
- 7 - AFC x
- 8 - MDS

- 9 - Clustering x
- 10 - Analyse de réseaux
- 11 - Analyse de variance et régression linéraire x
- 12 - Modèle linéaire généralisé x
- 13 - Modèles à décomposition d'erreur x
- 14 - Modèle d'équations structurelles (Lavaan)

- 15 - Times series
- 16 - Analyse spatiale et géographique
- 17 - Machine learning x

1.2 Les jeux de données

Au cours du développement, plusieurs cas pratiques - souvent réduit en volume pour rester exemplaire, seront employés. Les données seront partagées.

En voici la présentation des sets de données utilisées dans le syllabus. Elle sont disponible dans le répertoire "./data/"

- ESS : c'est une très belle base de données de sociologie.
- happydemics : observatoire de la présidentielle2022
- Arpur

1.3 Le cadre technique et les packages utilisés

Ce *syllabus* est écrit en **Markdown** (**?**) et avec le package **Bookdown** (**?**). Le code s'appuie sur `tidyverse` et emploie largement les ressources de `ggplot`. Les packages seront introduits au fur et à mesure. En voici la liste complète.

```
options(tinytex.verbose = TRUE)
knitr::opts_chunk$set(echo = TRUE, include=TRUE, cache=TRUE, message=FALSE, warning=FALSE)

#boite à outils et dataviz
library(tidyverse) # inclut ggplot pour la viz, readr et
library(cowplot) #pour créer des graphiques composés
library(ggridges) # le joy division touch
library(ggmosaic)
library(ggcorrplot)

#networks
library(igraph)
library(ggraph)

# Accéder aux données
library(rtweet)  # une interface efficace pour interroger l'api de Twitter

# NLP
library(tokenizers)
library(quanteda)
library(quanteda.textstats)
```

```
library(udpipe) #annotation syntaxique
library(tidytext)
library(cleanNLP) #annotation syntaxique

#sentiment
library(syuzhet)              #analyse du sentimeent



#mise en page des tableaux
library(flextable)


#statistiques et modèles
library(lme4) #pour des modèles plus complexe que les mco
library(jtools) #une série d'utiltaire pour bien représenter les résultats
library(interactions) #traitement des interactions
library(nlme)#pour les hlm
library(psych) #pour la psychometrie

#ACP et AFCM

library("FactoMineR")
library("factoextra")


#ML
library(caret)

#utilitaires
library(rcompanion)

#graphismes
library(ggthemes)
theme_set(theme_bw())


#palettes
library(colorspace) #pour les couleurs
library(wesanderson)

#regression
library(lme4)
library(jtools)
library(interactions)
```

```
library(betareg)


# Utilitaires

library(citr) #pour insérer des références dans le markdown

#config plot
theme_set(theme_minimal())
```

L'ensemble du code est disponible sur github. A ce stade c'est encore embryonnaire. Les proches et nos étudiants pourrons cependant y voir l'évolution du projet et de la progression

Quelques conventions d'écriture du code r

- On dénomme les dataframes de manière générale `df`, les tableaux intermédiaires sont appelé systématiquement `foo`
- Gestion des palettes de couleurs ** une couleur :" royalblue" ** deux couleurs ** 3 à 7 couleurs
- On emploie autant que possible le dialecte tidy.
- Les chunks sont notés en 4 chiffre : 2 pour le chapitre et deux pour le chunck. 0502 est le second chunk du chapitre 5.
- On commente au maximum les lignes de code pour épargner le corps du texte et le rendre lisible

1.4 A faire

todo list :

- insérer un compteur google analytics (voir https://stackoverflow.com/questions/41376989/how-to-include-google-analytics-in-an-rmarkdown-generated-github-page)
- modifier le titre en haut à gauche
- vérifier le système de références voir (https://doc.isara.fr/tuto-zothero-5-bibtex-rmarkdown-zotero/)
- Vérifier la publication en pdf

Chapter 2

Introduction aux data sciences

2.1 Objectif et sommaire

L'objet du manuel est de donner un aperçu général des méthodes d'analyses de données et de data science.

2.2 Science ou technique ?

Plûtôt que le terme consacré de Data sciences, il vaudrait mieux parler de data ingiénérie dans la mesure où le data scientiste participe à un processus de production qui va de l'acquisition des donnée à leur propagation dans l'organisation ou la société. La technique domine sur la science et l'unité se trouve dans l'intégration de ce processus. La révolution des données vient de l'interopérabilité croissante de ces techniques et d'une intégration qui fluidifie le passage d'une étape à une autre. Standards et langages en sont les éléments clés.

Du côté des sciences, ce dont bénéficie l'univers des data sciences, c'est l'héritage de cultures statistiques foisonnantes qui après s'être développées dans leur cocon displinaire, se retrouvent désormais rassemblées dans un même langage. Bien sur il y a de manière sous-jascente à ces cultures les mathématiques et les statistiques mathématiques qui construisent les fondements des modèles et des techniques. Mais le développement s'est fait souvent quand le scientifique se retrouve face à un problème où une observation.

Prenons le cas des psychologues qui ont inventé l'analyse factorielle dans le but de pouvoir tester certains de leurs concepts : un degré d'intelligence, une

personnalité, des attitudes.

Ou celui des écologues qui souhatent estimer une population de poisson dans une rivière, problème qui a donné naissance aux modèles de capture recapture. On pourrait ajouter les géographes avec les modèles d'analyse spatiale, les financiers face à la variabilité des cours des places boursières, etc. Celui des économètres est peut-être le plus évident. Les biostatisticiens sont des contributeurs importants.

Ce que la technique apporte c'est l'intégration par un langage et donc un ensemble de conventions, incarnées par r et python, d'algoritmes, et de programmes qui ne sont plus spécifique à un domaine, mais peuvent circuler de l'un à l'autre. C'est ainsi que le catalogues de toutes les techniques psychométriques devient accessible aux autres disciplines par le biais d'un pachage en particulier , psych. De la même manière l'outillage des linguiste devient accessible aux autres disciplines, pensons aux économiste qui intéègre dans le indicateurs des sources textuelle telle que l'analyse du sentiment.

L'interopérabilité apportée par ces langages ne se définit pas que par l'algorithme qui aurait été porté d'un autre langage vers celui-ci (des cas de réécriture ?) mais aussi par des programme passerelle qui à partir de r permettent d'activité des algorithme écrit en C, en javascript ou tout autre langage "plus informatiques" et souvent plus éfficace.

2.3 histoires des logiciels statistiques

Et c'est ce qu'on observe dans l'évolution des logiciels

- 1980 : statitcf
- 1980 : SAS comme accès à r
- 1990 : SPSS

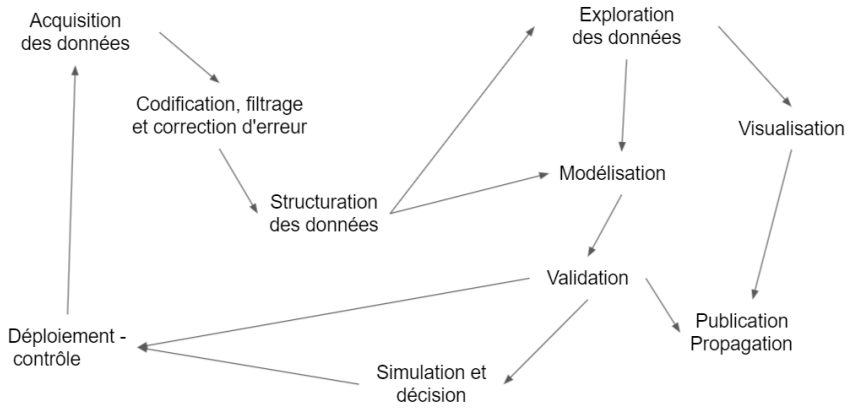
http://www.deenov.com/blog-deenov/histoire-du-logiciel-spad.aspx

des système portable

intégration graphique

la modularisation : base /fonction/ packages

2.4 Le processus de traitement des données



- Acquisition
- Codification , filtrage et correction d'erreur
- Structuration des données : api, open data
- Exploration
- Modélisation :
- validation : tests versus AB testing
- Simulation et décision
- Vizualisation et sensemaking
- Déploiement :
- Contrôle :
- Publication : dash board, pdf , slide etc, webb site

2.5 Les facteurs de développement des data-sciences

Ces développements sont favorisés par un environnement fertile dont trois facteurs se renforcent mutuellement.

2.5.1 Une lingua franca

histoire de r histoire de python

2.5.2 Une communauté

Le second facteur , intimement lié au premier, est la constitution d'une large communauté de développeurs et d'utilisateurs qui se retrouvent aujourd'hui

dans des plateformes de dépots (Github, Gitlab), de plateformes de type quora (StalkOverFlow), de tutoriaux, de blogs (BloggeR), de journaux (Journal of Statistical Software) et de bookdown.

Des ressources abondantes sont ainsi disponibles et facilitent la formation des chercheurs et des data scientists. Toutes les conditions sont réunies pour engendrer une effervescence créative.

2.5.3 La multiplication des sources de données.

Le troisième est la multiplication des sources de données et leur facilité d'accès. Les données privées, et en particulier celles des réseaux sociaux, même si un péage doit être payé pour accéder aux APIs, popularisent le traitement de données massives. Le mouvement des données ouvertes (open data) proposent et facilitent l'accès à des milliers de corps de données : retards de la SNCF, grand débat, le formidable travail de l'Insee, european survey etc.

2.5.4 du ML à l'IA

Le retour au boites noires dans les années 2000. Ce qui distingue les statistiques traditionnelles de l'approche machine learning réside d'abord par une approche de la modélisation différente.

Les modèles statistiques et économétriques considèrent une structure de relation, la spécification du modèle (ex : le modèle linéaire), mais aussi des modèles de distribution des erreurs qui définissent le cadre d'estimation. L'évaluation passe par le test des hypothèses sur les paramètres de la qualité d'ajustement.

Le machine learning, se concentre sur la valeur prédictive, et considère n'importe quelle spécification même si elle est peu intelligible et comprend de grandes quantités de paramètres sur lesquels aucun test n'est produit.

Les deux approches ont plutôt tendance à ce compléter, les première testant des théories, les secondes procurant aux première de nouvelles hypothèse par de nouvelle mesure. Pour en donner un exemple simple, l'analyse de sentiment emploie des modèles complexes pour le prédire avec le seul texte, l'IA permet d'enrichir des données empiriques par exemple en testant en finance la relation de cet indicateurs aux prix de marché. Un autre exemple en marketing.

Les méthodes disponible se sont accumulée depuis ces derbière 20 années

- 1960

KNN, SVM, rf et le retour des réseaux de neurones.

La révolution des convolutions et la multiplication des architectures

Chapter 3

Prise en main de r

Pour démarrer :

- 1 - Télécharger et installer r sur le site du Comprehensive r Archive Network
- 2 - Télécharger et installer Rstudio.(version free)
- 3 - Dans le cadre de cet atelier, on adopte la méthode du rmarkdown. On recommande fortement de lire l'ouvrage de référence, même si la prise en main est très rapide.
- 4 - Il est désormais indispensable d'utiliser le package `tidyverse` et en particulier les fonctions de manipulation et de pipe (%>%) fournies par `dplyr`. Ce sera donc le premier package à installer (attention, il appele de nombreuses dépendences, l'installation peut prendre plusieurs minutes)

3.1 La convention du Rmarkdown

Différentes manières d'interagir avec r sont possibles : la première est le mode console, pour de petite opérations et un utilisateur chevronné, celà peut être commode car rapide mais très rapidement on sera amené à enregistrer les opérations dans des scripts. Une idée novatrice a été d'intégrer l'ensemble des éléments dans un seul document : le script découpé en petits éléments : des chunks, le commentaire et l'analyse verbabe dans un format texte, et le résultat. Dans l'univers python il s'agit des carnets Jupiter, pour r c'est le rmarkdown.

C'est un dialecte du markdown générique adapté au langage r. On recommande au lecteur d'en lire le manuel et de le garder dans ses onglets.

Quelques éléments de base :

un document markdown est composé de plusieurs éléments

1. Yalm dans cet entête les éléments essentiels sont définis et paramétrés
2. Texte : il suit les conventions de mise en forme du html :
    - des # pour les niveau de titres
    - (x) [html] pour des liens et [.jpg] ()" pour des images
3. Les chunks sont isolés par 3 tiks au début et à la fin.
4. Résultats apparaissent sous les chunks

Ce document peut être exécuté et publié sous différents formats : html, pdf ou même word avec les éléments suivants:

- plan
- texte
- code
- résultats
- Bibliographie
- Références
- liens
- images

3.2 Lire les données

La première étape c'est la lecture des données. On commence par le plus simple la lecture de fichiers locaux, dont les formats sont multiples : csv, tsv, xlsx, Spss, etc… Le package `readr` contribue à cette tâche.

```
df <- read_csv("./Data/BXL_listings.csv")
```

Il est possible aussi d'accéder en direct aux données du web, c'est bien utile pour s'assurer que les données sont bien fraîches. Par exemple une connexion à Nsppolls qui propose une compilation de tous les sondages d'intention de vote de la présidentielle 2022.

```
df_pol <- read_delim("https://raw.githubusercontent.com/nsppolls/nsppolls/master/presi
                     delim = ",", escape_double = FALSE, trim_ws = TRUE)
```

3.2.1 La diversité des formats

Peu de formats échappent à r, ils peuvent faire appel à des packages spécifiques

- excell
- Json
- shape et autre GIS :
- les formats bibliographique : bib et ris

3.3 Dplyr pour manipuler les données

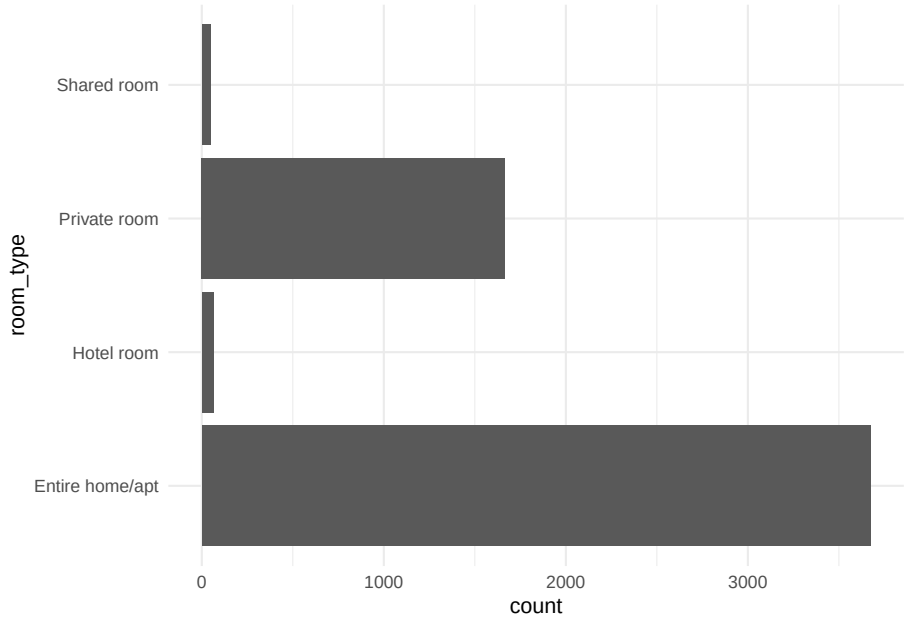
C'est un des packages essentiels de la suite tidyverse. Il permet de manipuler aisément les données et mérite une étude approfondie. Un point de départ ou en français : dplyr .

3.3.1 Des pipes %>%

Une grand part de l'intérêt de dplyr est de reprendre un opérateur de magritr très utile : le pipe noté `%>%`. Celui ci permet de passer le résultat de l'opération à gauche, dans la fonction à droite.

Un exemple simple : dans la ligne de code suivante, une première fonction lit le fichier CSV, et envoie le résultat de cette lecture dans une fonction graphique élémentaire: compter les occurrences des modalités de la variable room_type.

```
g <- read_csv("./Data/BXL_listings.csv") %>%
  ggplot(aes(x=room_type))+
  geom_bar()+
  coord_flip()
g
```



3.3.2 Des verbes

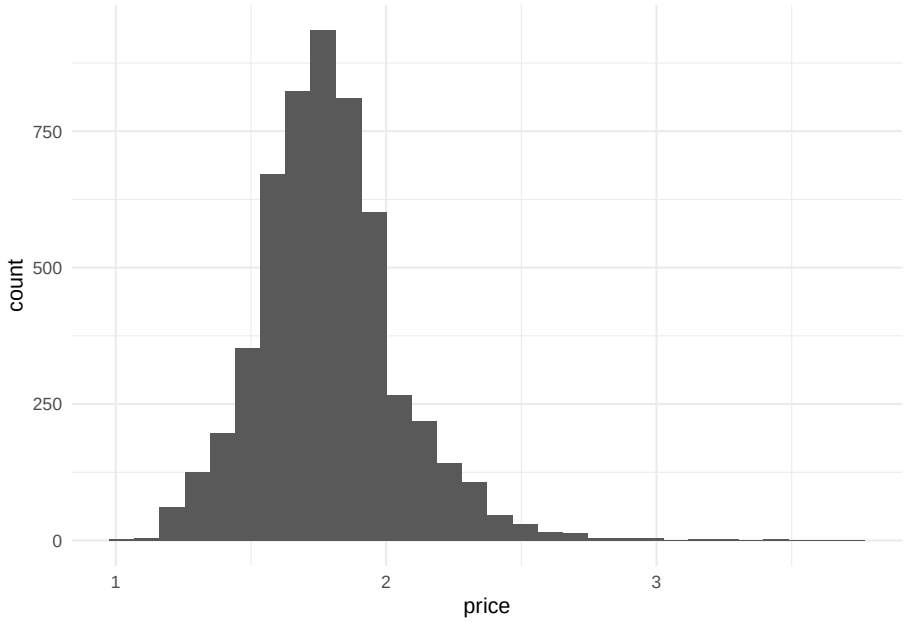
L'originalité de dplyr est de définir des fonctions comme des verbes. Chaque verbe désigne un type d'action. On va les examiner progressivement. Ils sont simples à comprendre : transformer une variable, filtrer les observations selon un critère, isoler des variables, les grouper pour en calculer des résultats statistiques (somme, moyenne, variance, max min etc), les déployer selon un format long ou les distribuer en différents critères, les fusionner enfin.

3.3.2.1 Mutate

En Français c'est "transformer". On modifie la valeur d'une variable par une fonction plus ou moins complexe, éventuellement en ajoutant des conditions.

Dans notre exemple, faisant au plus simple, puisque la distribution est asymétrique, une transformation du prix par les log peut donner des résultats intéressants. Et c'est le cas. On retrouve une distribution qui semble être gaussienne.

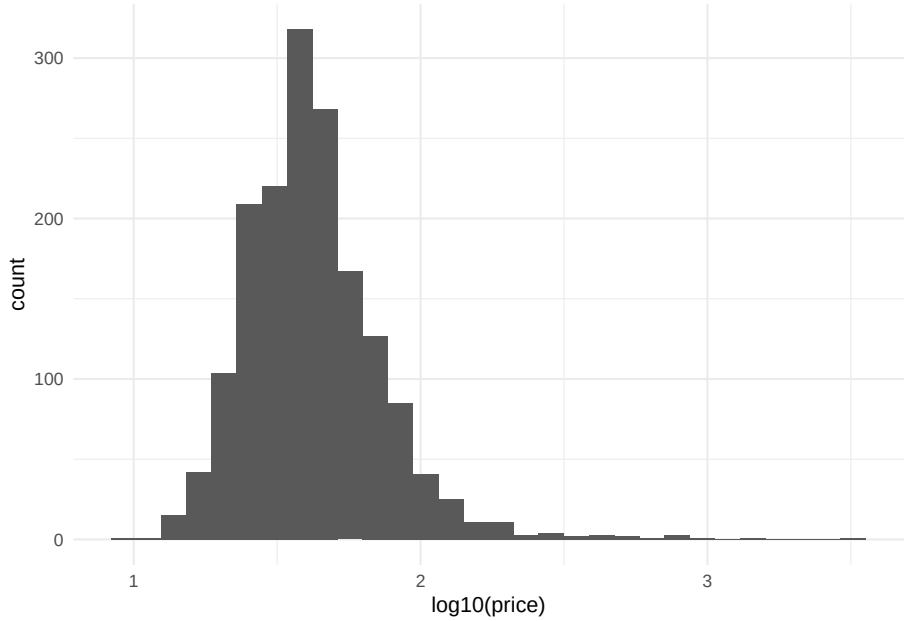
```
g <- read_csv("./Data/BXL_listings.csv") %>%
  mutate(price=log10(price))%>%
  ggplot(aes(x=price))+
  geom_histogram()
g
```



3.3.2.2 Filter

On peut vouloir se concentrer sur une sous population. Par exemple les chambres privées.

```
g <- read_csv("./Data/BXL_listings.csv") %>%
  filter(room_type=="Private room" ) %>%
    mutate(price=price)%>%
  ggplot(aes(x=log10(price)))+
  geom_histogram()
g
```



3.3.2.3 select

On peut sélectionner des colonnes pour créer un tableau spécifique. On en profite pour introduire 'flextable' , une solution élégante pour éditer des tableaux en html.

```
foo <- read_csv("./Data/BXL_listings.csv") %>%
  dplyr::select(room_type,price)

ft <- flextable(foo[ sample.int(10),])%>%
   set_header_labels(room_type="Type de logement",
```

```
  price = "Prix en euros")%>%
  theme_vanilla()%>%
  autofit()
ft
```

Type de logement	Prix en euros
Entire home/apt	80
Entire home/apt	80
Entire home/apt	74
Entire home/apt	91
Hotel room	120
Entire home/apt	85
Entire home/apt	65
Entire home/apt	95
Entire home/apt	200
Entire home/apt	74

3.3.2.4 Group_by et summarize

c'est une opération clé, en groupant selon les modalités d'une ou plusieurs variables, on peut construire des tableaux agrégés. On l'associera à `summarize` qui permet de calculer les statistiques agrégées selon le groupe que l'on a définit.

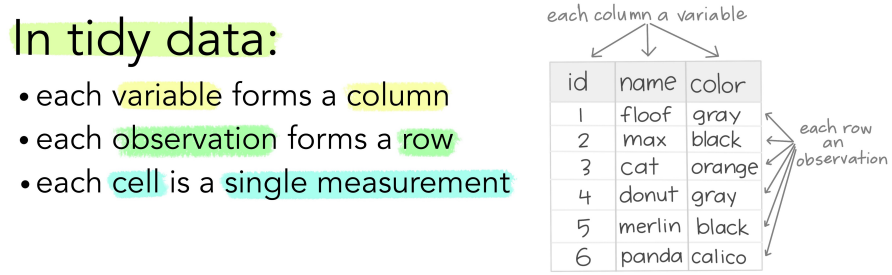
```
foo <- read_csv("./Data/BXL_listings.csv")%>%
  dplyr::select(neighbourhood, price)%>%
    group_by(neighbourhood ) %>%
  summarise(averageprice=round(mean(price),2),
            nombreoffre=n())

ft <- flextable(foo)%>%
     set_header_labels(neighbourhood="Type de logement",
  averageprice = "Prix en euros",
  nombreoffre="Nombre d'offre")%>%
  theme_vanilla()

ft
```

Type de logement	Prix en euros	Nombre d'offre
Anderlecht	71.89	232
Auderghem	66.27	77
Berchem-Sainte-Agathe	65.87	31
Bruxelles	91.02	1,759
Etterbeek	75.76	296
Evere	70.00	41
Forest	64.89	226
Ganshoren	50.48	21
Ixelles	81.46	849
Jette	70.29	75
Koekelberg	70.73	37
Molenbeek-Saint-Jean	67.42	179
Saint-Gilles	76.08	589
Saint-Josse-ten-Noode	55.20	136
Schaerbeek	61.90	364
Uccle	75.50	274
Watermael-Boitsfort	74.80	65
Woluwe-Saint-Lambert	62.47	121
Woluwe-Saint-Pierre	110.99	81

3.3.2.5 Pivot_wider et pivot_longer

Si pour l'habitué des feuilles de calculs les données croisent des observations avec des variables, ce format n'est pas le seul moyen de représenter des données, et pas forcément le meilleur.

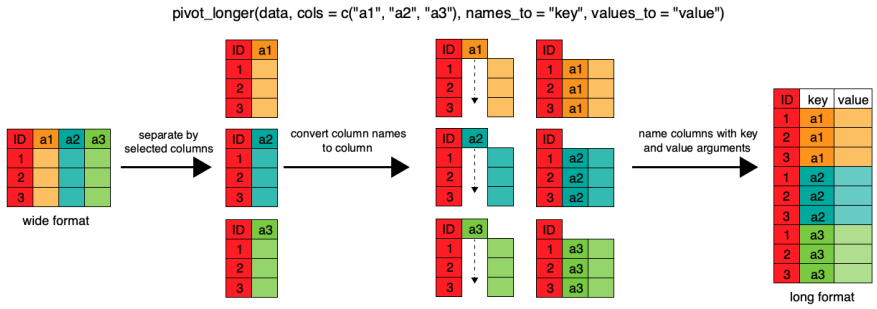
Une théorie des tidy data a été proposé par wickham : Un ensemble de données est une collection de valeurs, généralement des nombres (si elles sont quantitatives) ou des chaînes de caractères (si elles sont qualitatives). Les valeurs sont organisées de deux manières. Chaque valeur appartient à une variable et à une observation. Une variable contient toutes les valeurs qui mesurent le même attribut sous-jacent (comme la hauteur, la température, la durée) dans différentes unités. Une observation contient toutes les valeurs mesurées sur la même unité (comme une personne, ou un jour, ou une course) à travers les attributs.



Wickham, H. (2014). Tidy Data. Journal of Statistical Software 59 (10). DOI: 10.18637/jss.v059.i10

Figure 3.1: merge

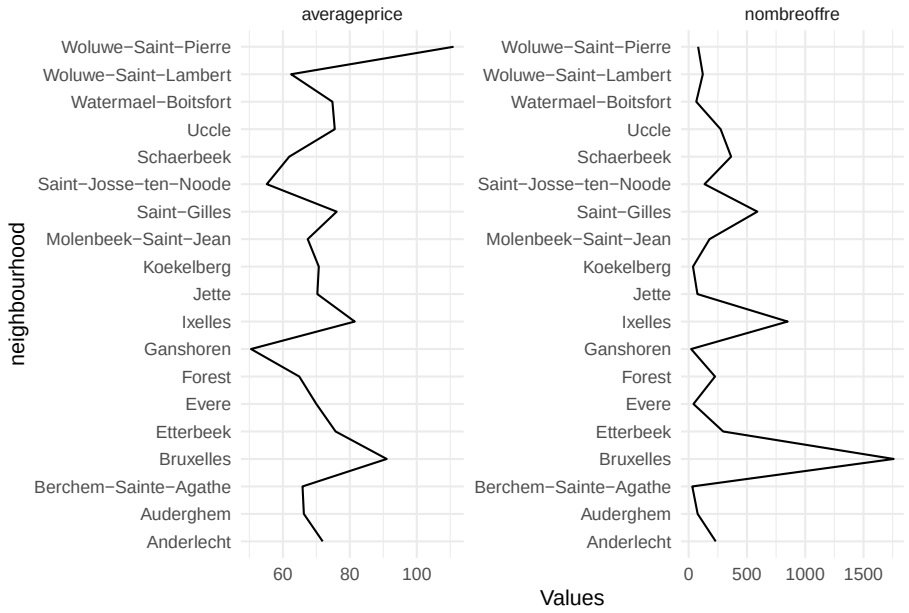
Pour passer d'un tableau individu x variable à une structure ordonnée, la fonction pivot_longer est particulièrement appropriée. En voici l'anatomie.



Et un exemple numérique :

```
foo <- foo %>%
  pivot_longer(-neighbourhood,names_to= "Variables",values_to = "Values")
```

```
ggplot(foo, aes(x=neighbourhood, y=Values, group=Variables))+
  geom_line()+facet_wrap(vars(Variables),scales="free")+
  coord_flip()
```



L'opération inverste est de partir d'un tableau long vers un tableau large.

country	year	cases
Angola	1999	800
Angola	2000	750
Angola	2001	925
Angola	2002	1020
India	1999	20100
India	2000	25650
India	2001	26800
India	2002	27255
Mongolia	1999	450
Mongolia	2000	512
Mongolia	2001	510
Mongolia	2002	586

country	1999	2000	2001	2002
Angola	800	750	925	1020
India	20100	25650	26800	27255
Mongolia	450	512	510	586

Pivot data wider

```
data %>%
  pivot_wider(
    names_from = "year",
    values_from = "cases"
  )
```

On remarquera que l'usage de cette fonction est nécessaire dans l'emploi de ggplot qui suit la logique des tidy data, ou données ordonnées

3.3.3 Fusionner les données

On sera souvent amené à construire des tableaux de données en les enrichissant par d'autres tableaux et à fusionner les données.

Le cas le plus simple est d'ajouter d'autres observations à un fichier de données. On distingue deux cas :

- les deux tableaux concernent les mêmes individus classé dans le même ordre, seules les colonnes diffèrent. On utilisera la fonction `cbind()`
- si les variables sont identiques mais que les individus sont différents on peut concatène des données avec `rbind()` (L'équivalent de DPLYR est row_bind et column_bind)

```
x1<-as.data.frame(c(1,2,3,4,5)) %>%rename(x=1)

y<-as.data.frame(c("a","b","c","d","e"))  %>%rename(y=1)

z<-cbind(x1,y)

ft<-flextable(z)
ft
```

x	y
1	a
2	b
3	c
4	d
5	e

```
x2<-as.data.frame(c(9,8,7,6)) %>%rename(x=1)

w<-rbind(x1,x2)

ft<-flextable(w)
ft
```

x
1

x
2
3
4
5
9
8
7
6

mais très souvent on sera dans des cas différents et la fusion des données devra suivre des index

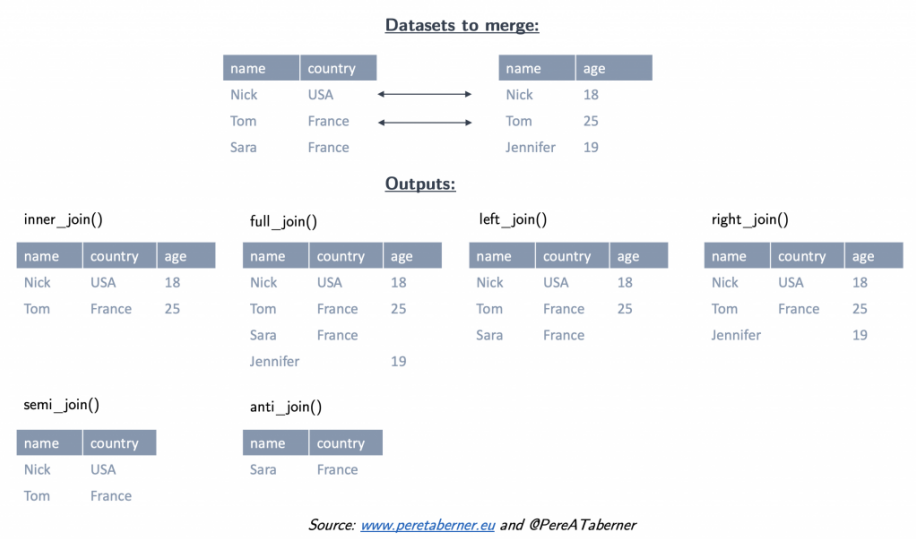


Figure 3.2: merge

Plusieurs types de fusion sont proposées.

générale

fusion à gauche

fusion à droite

(Voir ce cours)[https://coletl.github.io/tidy_intro/lessons/dplyr_join/dplyr_join.html]

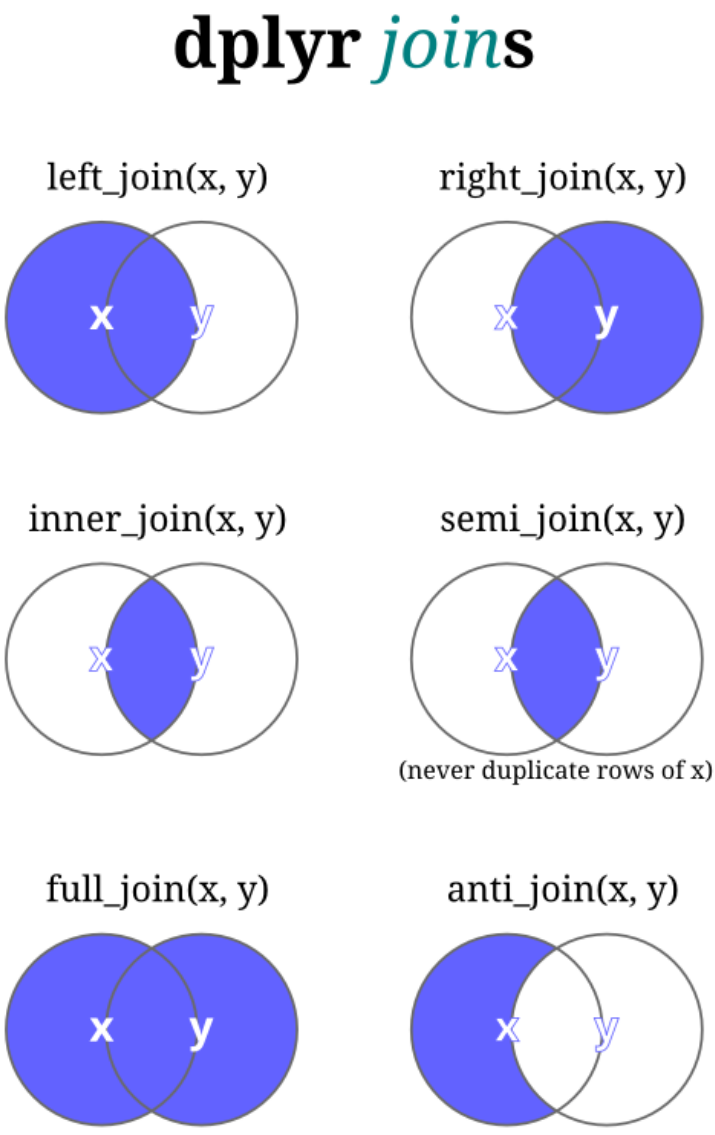


Figure 3.3: Mode de fusions

Chapter 4

Analyses univariées

Nous avons appris à lire des données, à les manipuler, nous avons le droit d'être pressé de les représenter de manière immédiatement lisible, par des dataviz.

On présente d'abord rapidement le concept de grammaire des graphiques

On se concentre ensuite sur un cas d'étude

On décline.

4.1 La grammaire des graphiques

C'est sans doute une des percées conceptuelles laplus intéressante des data-sciences. La représentation graphhiques des données fait l'objet à la fois d'une explosion créative mais aussi d'une synthèse théorique. C'est l'apport de la grammaire des graphiques.

Ces outils s'appuient sur l'idée de grammaire des graphiques. En voici un clair résumé.En français il y a toujours le larmarange

4.1.1 Un modèle en couche

Celle-ci met un ordre dans les éléments qui composent un graphique et les superpose.

- l'aesthetic definit les éléments que l'on veut représenter : ce qu'on met en abscisse, ce qu'on met en ordonnné, les groupes que l'on veut distinguer.
- la geométrie (geom_x)qui définit la forme de représentation
- les échelles (scale_x)
- Labelisation (labs)

Figure 4.1: layers

- les templates
- le facetting

ggplot est construit selon cette structure. Voici le book de référence, qui est au centre de ce cours. On aura besoin de manière assez systématique de manipuler les données avant de les représenter, dplyr nous permet de le faire aisément.

4.1.2 Une typologie des représentations

Un point de départ fondamental est la gallery de ggplot„ elle présente de manière synthétique la plupart les types de figures qui peuvent être représentées, avec du code facilement reproductible.

Une classification simple

- Analyse univariée
- Analyse bi variée
- Analyse multivariée ** les variables sont quantitatives : on analyse des matrices de corrélations ** les variables sont qualitatives : on analyse des tableaux croisés
- Analyse geospatiale
- Analyse de réseaux
- analyse d'arbres
- Diagramme de flux

4.1.3 L'esthétique

L'art des couleurs tient dans les palettes on aimera celles de Wes Anderson, on peut adorer fishualize. on trouvera

4.2 Une étude de cas

Les données sont extraites de l'ESS, une sélection est disponible ici. Elle couvre les 9 vagues et concernent la France et L'Allemagne. Les variables dépendantes (celles que l'on veut étudier et expliquer) sont les 9 items de la confiance, les variable considérées comme indépendantes (ou explicatives) sont une sélection de variables socio-démographiques : âge, genre, perception du pouvoir d'achat, orientation politique, type d'habitat.

On fait quelques opérations de recodage et on renomme les variables avoir une lecture plus aisée des variables et de leurs catégories. Le plan de recodage d'un jeu de données qu'on va employer dans les chapitres suivants. Il s'appuie sur le langage de base.

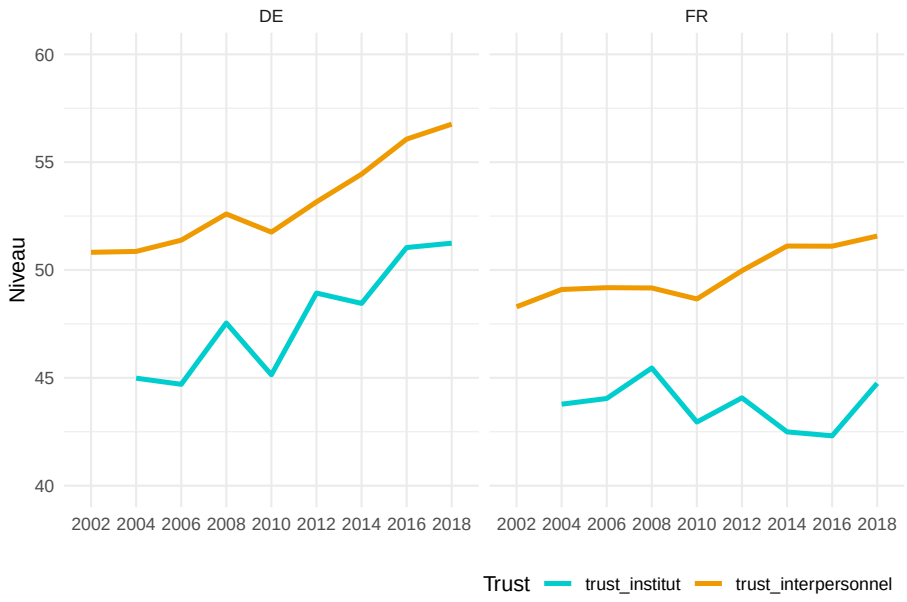
L'analyse univarié, comme son nom l'indique, ne s'intéresse qu'à une seule variable. Celle-ci peut être **quantitative** ou **qualitative** etne comporter qu'un nombre limité de modalités entre lesquels aucune comparaison de grandeur ne peut être faite. Les premières ont le plus souvent dans r un format numeric, les autres correspondent au format *factor*.

(Un exercice peut être de le réécrire avec dplyr.)

```
df<-readRDS("./data/trustFrAll.rds")

#quelques recodages
#on renomme pour plus de clarte
names(df)[names(df)=="trstun"] <- "NationsUnies"
names(df)[names(df)=="trstep"] <- "ParlementEurop"
names(df)[names(df)=="trstlgl"] <- "Justice"
names(df)[names(df)=="trstplc"] <- "Police"
names(df)[names(df)=="trstplt"] <- "Politiques"
names(df)[names(df)=="trstprl"] <-"Parlement"
names(df)[names(df)=="trstprt"] <- "Partis"
names(df)[names(df)=="pplhlp"] <- "help"
names(df)[names(df)=="pplfair"] <- "fair"
names(df)[names(df)=="ppltrst"] <- "trust"

#on construit les scores de confiance
df<-df %>%
  mutate(trust_institut=(Partis+Parlement+Politiques+Police+Justice+NationsUnies+ParlementEurop)*
```

```
df$Year<-2000
#recodage des variables independantes
df$Year[df$essround==1]<-2002
df$Year[df$essround==2]<-2004
df$Year[df$essround==3]<-2006
df$Year[df$essround==4]<-2008
df$Year[df$essround==5]<-2010
df$Year[df$essround==6]<-2012
df$Year[df$essround==7]<-2014
df$Year[df$essround==8]<-2016
df$Year[df$essround==9]<-2018
df$Year<-as.factor(df$Year)

df$OP<-" "
#ggplot(df,aes(x=lrscale))+geom_histogram()
df$OP[df$lrscale==0] <- "Extrême gauche"
df$OP[df$lrscale==1] <- "Gauche"
df$OP[df$lrscale==2] <- "Gauche"
df$OP[df$lrscale==3] <- "Centre Gauche"
df$OP[df$lrscale==4] <- "Centre Gauche"
df$OP[df$lrscale==5] <- "Ni G ni D"
df$OP[df$lrscale==6] <- "Centre Droit"
df$OP[df$lrscale==7] <- "Centre Droit"
df$OP[df$lrscale==8] <- "Droite"
df$OP[df$lrscale==9] <- "Droite"
df$OP[df$lrscale==10] <- "Extrême droite"
#la ligne suivante est pour ordonner les modalités de la variables
df$OP<-factor(df$OP,levels=c("Extrême droite","Droite","Centre Droit","Ni G ni D","Cent


df$revenu<-" "
df$revenu[df$hincfel>4] <- NA
df$revenu[df$hincfel==1] <- "Vie confortable"
df$revenu[df$hincfel==2] <- "Se débrouille avec son revenu"
df$revenu[df$hincfel==3] <- "Revenu insuffisant"
df$revenu[df$hincfel==4] <- "Revenu très insuffisant"
df$revenu<-factor(df$revenu,levels=c("Vie confortable","Se débrouille avec son revenu"

df$habitat<-" "

df$habitat[df$domicil==1]<- "Big city"
df$habitat[df$domicil==2]<-"Suburbs"
df$habitat[df$domicil==3]<-"Town"
df$habitat[df$domicil==4]<-"Village"
df$habitat[df$domicil==5]<-"Countryside"
```

```
df$habitat<-factor(df$habitat,levels=c("Big city","Suburbs","Town","Village","Countryside"))

df$genre<-" "

df$genre[df$gndr==1]<-"H"
df$genre[df$gndr==2]<-"F"

df$age<-" "

df$age[df$agea<26]<-"25<"
df$age[df$agea>25 & df$agea<36]<-"26-35"
df$age[df$agea>35 & df$agea<46]<-"36-45"
df$age[df$agea>45 & df$agea<66]<-"46-65"
df$age[df$agea>65 & df$agea<76]<-"66-75"
df$age[df$agea>75]<-"75>"
df$age<-factor(df$age,levels=c("25<","26-35","36-45","46-65","66-75", "75>"))

saveRDS(df, "./data/dfTrust.rds)")

foo<-df%>%
  dplyr::select(Year,cntry, trust_institut, trust_interpersonnel)%>%
  group_by(Year,cntry)%>%
  summarise(trust_institut=mean(trust_institut, na.rm=TRUE),
            trust_interpersonnel=mean(trust_interpersonnel, na.rm=TRUE))
foo$Year<- as.character(foo$Year)
foo$cntry<- as.character(foo$cntry)

foo<-foo%>%pivot_longer(!c(Year,cntry),names_to="Trust", values_to="value" )


ggplot(foo,aes(x=Year, y=value, group=Trust))+
  geom_line(stat="identity",aes(color=Trust), size=1.2)+
  facet_wrap(vars(cntry))+
  scale_color_manual(values = c("Cyan3","Orange2"))+ theme(
    legend.position = "bottom",
    legend.justification = c("right", "top"),
    legend.box.just = "right",
    legend.margin = margin(6, 6, 6, 6)
    )+ labs(x=NULL, y="Niveau")+ylim(40,60)
```



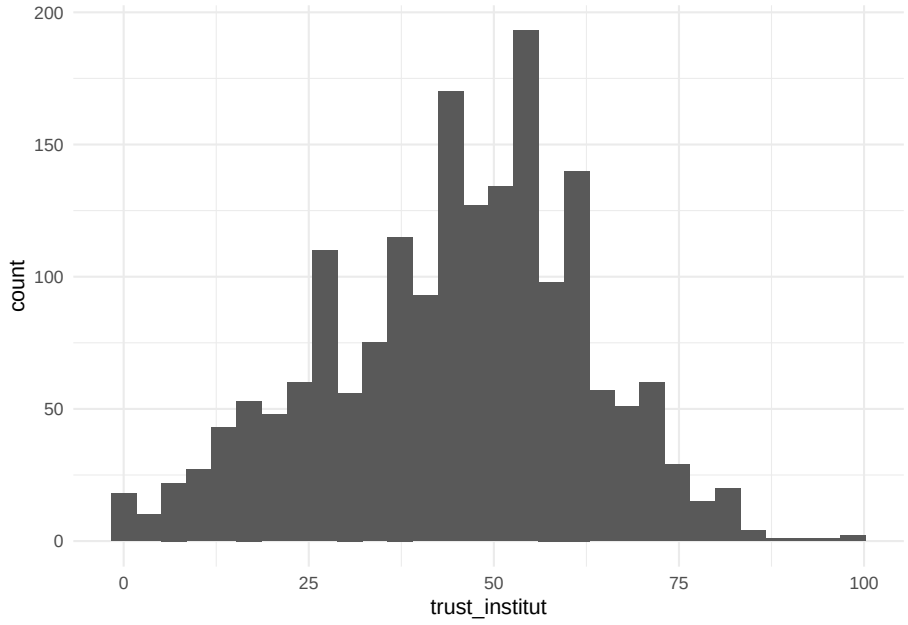
4.2.1 Le cas des variables quantitatives

Les variables quantitatives décrivent une variable dont les valeurs décrivent les quantités d'une grandeur. Elle peuvent être discrètes (dénombrement du d'un nombre d'unités) - le nombre d'habitant), ou continue (le nombre de km parcourus). l'**histogramme** est l'outil de base pour représenter la distribution d'une telle variable. Il représente pour des intervalles de valeurs donnés, la fréquence des observations.

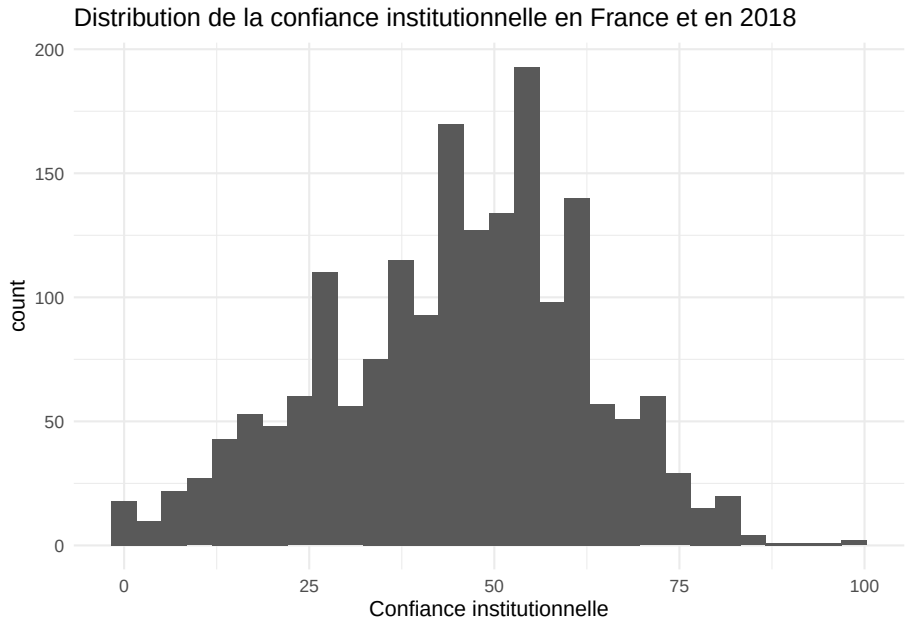
Sa syntaxe simple comporte d'abord la définition de la variable et de la source de données, puis une des "géométrie" de ggplot : la fonction geom_histogram. Dans notre exemple, on va représenter le score de confiance institutionnelle pour la France en se concentrant sur la dernière vague d'enquête.

```
#On charge le fichier recodé à la fin du chapitre précédent
df<-readRDS("./data/dfTrust.rds)")

#filtrage sur 2018 et la France.

foo<-df%>%
  filter(Year=="2018" & cntry=="FR" & !is.na(trust_institut))

# on stocke le diagramme dans l'objet g00, pour le réutiliser ultérieurement et pouvoi
g00<-ggplot(foo,aes(x=trust_institut))+
```

```
  geom_histogram()

g00
```




```
g00+labs(title="Distribution de la confiance institutionnelle en France et en 2018",
         x="Confiance institutionnelle")
```



On va améliorer l'aspect en

a) modifiant la couleur et la largeur des barres,
b) ajoutant un thème,
c) en précisant les éléments textuels (titres, label)
d) en calculant et en représentant la valeur moyenne et l'écart-type . Pour ces statistiques, on emploie les fonction de base : mean, sd et round.

On notera que le titre est défini par la concaténation de plusieurs chaines de caractères avec la fonction paste0. On peut ainsi injecter dans le graphique des éléments externes au jeu de données.

```
#on calcule la moyenne
moy=mean(foo$trust_institut, na.rm=TRUE)
sd=sd(foo$trust_institut, na.rm=TRUE)

#avec tous les éléments
g01 <-ggplot(foo,aes(x=trust_institut))+
  geom_histogram(binwidth=5,fill="pink")+
  labs(title= "Distribution de la confiance institutionnelle",
       subtitle= paste0("moyenne = ",round(moy,2), " ecart-type = ", round(sd,2)),
       caption="ESS2002-2018",
       y= "frequence",
       x="confiance (index de 0 à 100)")+
    geom_vline(xintercept=moy, color="red",size=1.5)+
```

```
        geom_segment(y = 0, yend=0,x=moy-sd,xend=moy+sd, color="orange",size=1.5)

g01
```

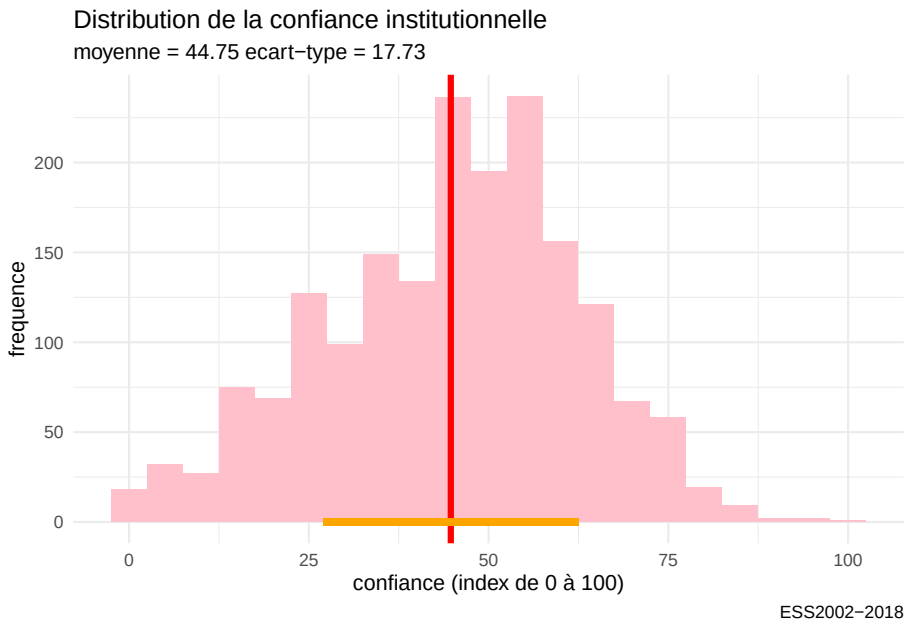
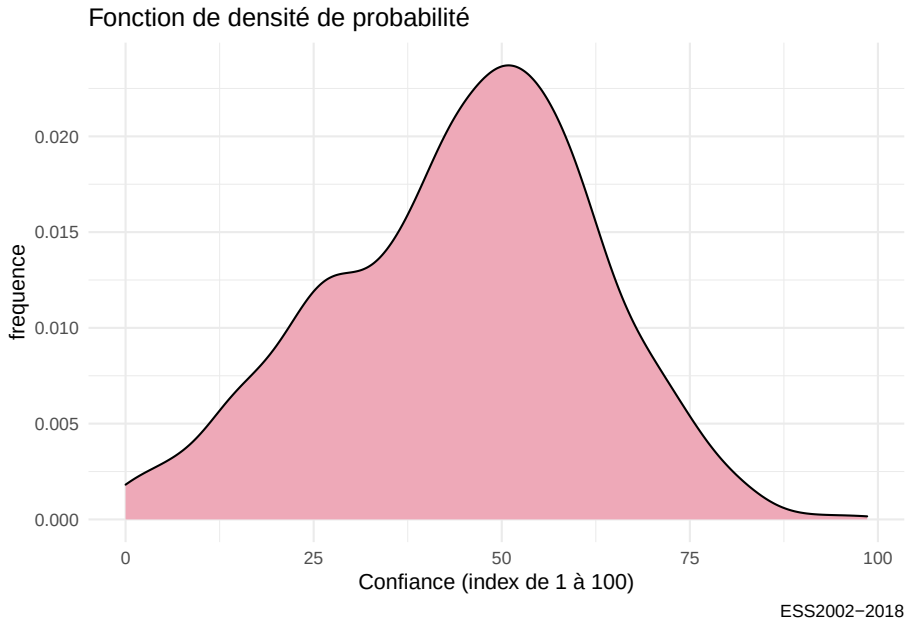


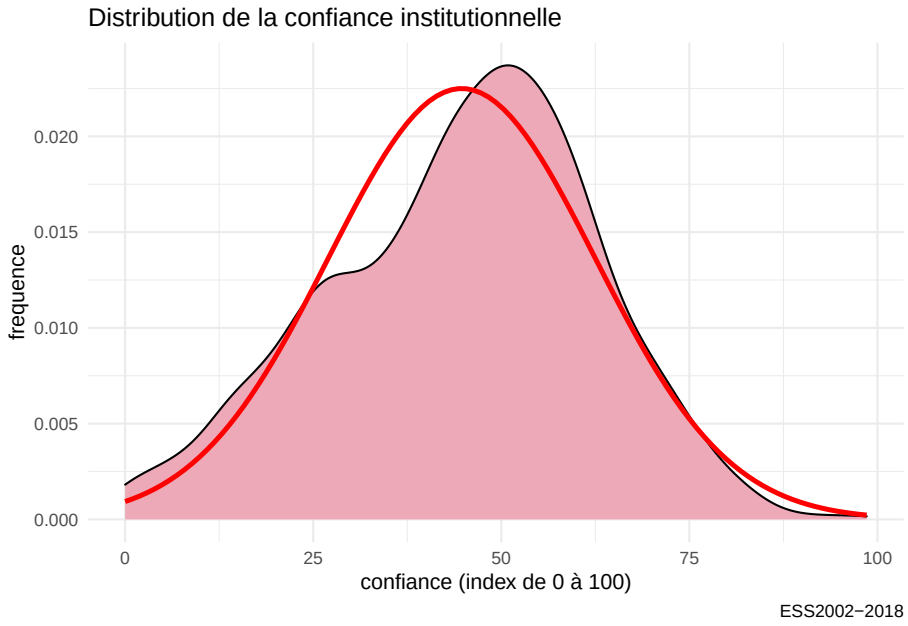
Diagramme de densité : Au lieu de représenter les effectifs, on ramène l'effectif total à 1.

```
g04<-ggplot(foo,aes(x=trust_institut))+
  geom_density(fill="pink2") +
  labs(title= "Fonction de densité de probabilité", caption="ESS2002-2018",y= "frequence",x="Conf
g04
```



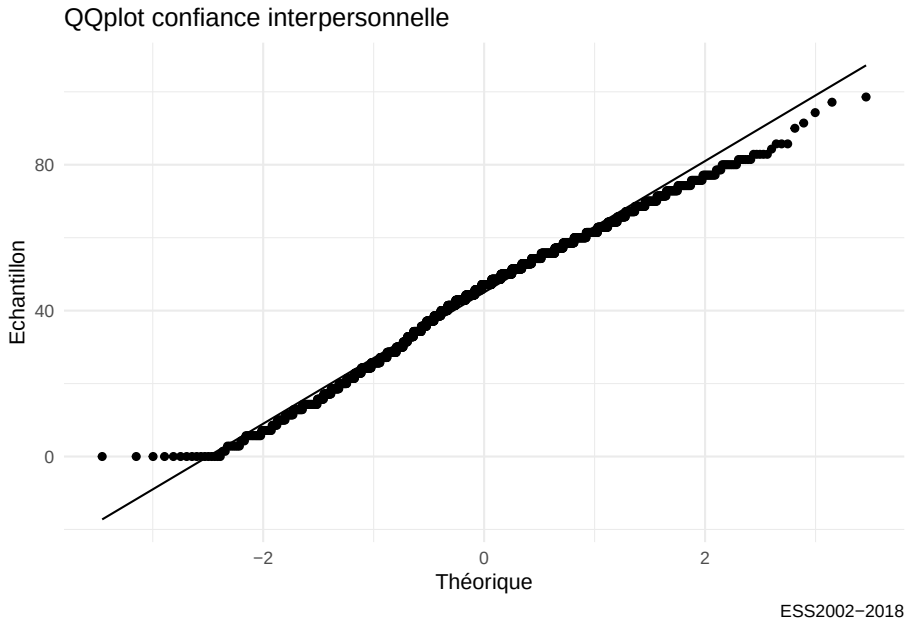
enfin on peut examiner par rapport à une distribution théorique, en l'occurrence une distribution gaussienne, ou normale, de paramètres égaux à la moyenne et la variance empirique de la distribution. L'ajustement est convenable même si on observe une déviation sur la droite. C'est pourquoi on calcule aussi la Kurtosis et le skewness de la distribution.

```
#On a déjà calculé la moyenne : mean
#il nous manque l'écart-type et
sd<-sd(foo$trust_institut, na.rm=TRUE)
library(moments)
sk<-skewness(foo$trust_institut)
ks<-kurtosis(foo$trust_institut)


g05<-ggplot(foo,aes(x=trust_institut))+
  labs(title= "Distribution de la confiance institutionnelle", caption="ESS2002-2018",y
  geom_density(fill="pink2")+
  stat_function(fun = dnorm,color="red",size=1.2, args = list(mean =moy, sd=sd))

g05
```



Un grand classique du test de normalité d'une distribution est le diagramme QQ

```
g06 <- ggplot(foo, aes(sample = trust_institut)) +
  stat_qq() + stat_qq_line()+
  labs(title= "QQplot confiance interpersonnelle", caption="ESS2002-2018",y= "Echantillon",x="Thé
g06
```



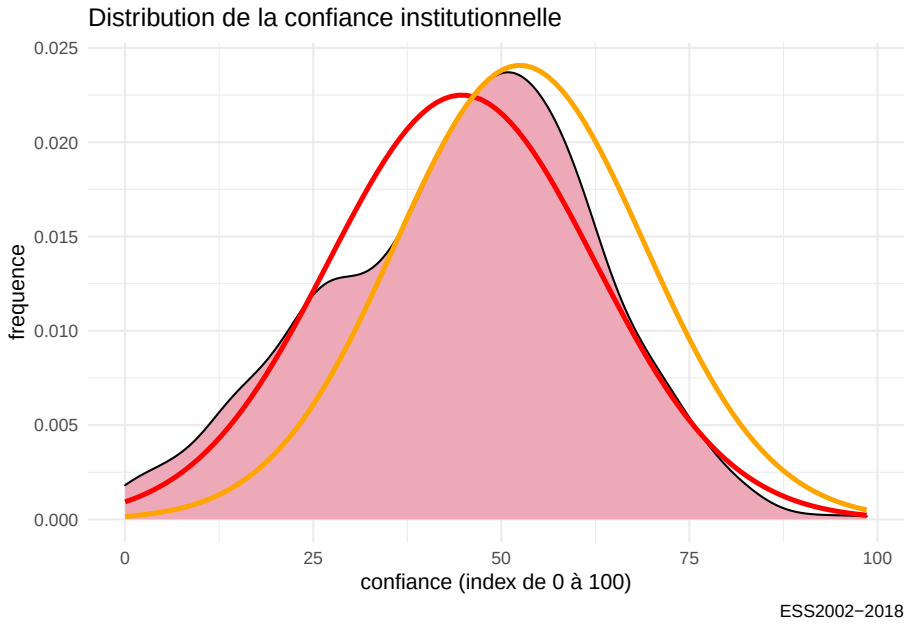
On fini cette étude détaillée par l'ajustement d'abord d'un modèle (loi normale) aux données. Ensuite d'un modèle de mélange (Mixture model) par lequel on défiit la loi de distribution sous jascente, comme un mélange entre deux populations normale de paramètres distincts.

https://tinyheero.github.io/2015/10/13/mixture-model.html

```
df0<-df %>% na.omit()
library(MASS)
fit<-fitdistr(df0$trust_interpersonnel,"normal")
fit
```

```
##       mean           sd
##   52.48548790   16.57617220
##  ( 0.09344363) ( 0.06607462)
```

```
g07<- g05+stat_function(fun =  dnorm ,color="orange",size=1.2, args = list( mean=52.48
g07
```




```
library(mixtools)
trust = foo$trust_institut
mixmdl = normalmixEM(trust, k=2)
```

```
## number of iterations= 271
```

```
mixmdl$mu
```

```
## [1] 20.77246 50.86797
```

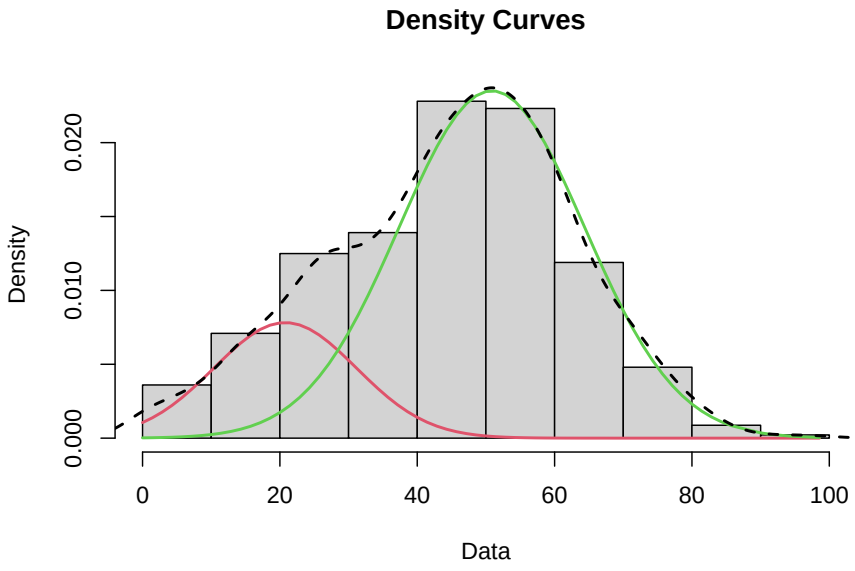
```
mixmdl$sigma
```

```
## [1] 10.37739 13.51802
```

```
mixmdl$lambda
```

```
## [1] 0.2034332 0.7965668
```

```
plot(mixmdl,which=2)
lines(density(trust), lty=2, lwd=2)
```

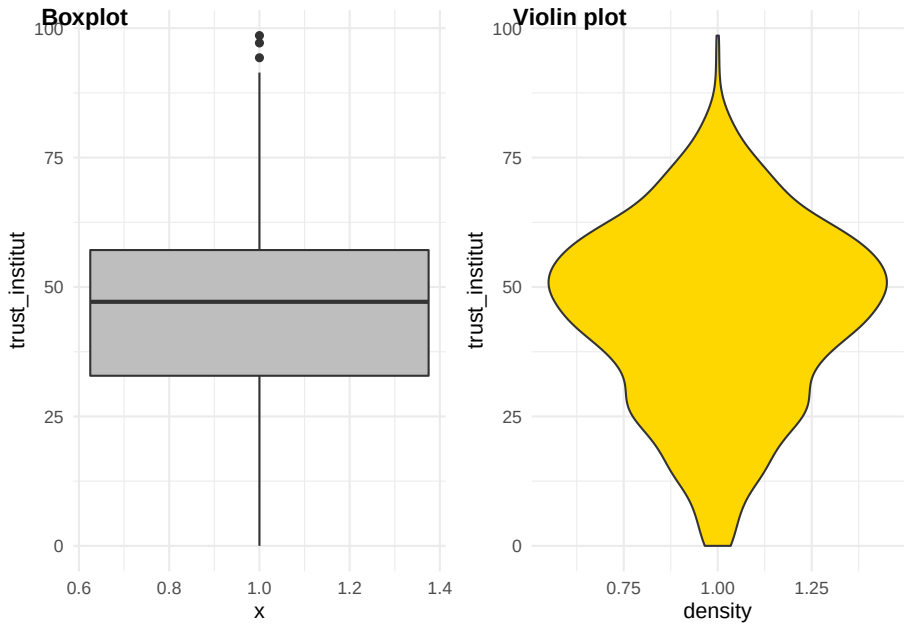


Finalement si notre distribution est univariée, car n'étudiant qu'une variable, on peut quand distinguer deux population distinctes.

4.2.1.1 D'autres méthodes

Il n'y a pas que l'histogram ou le diagramme de densité, d'autres méthodes sont utiles, surtout quand on veut comparer des groupes (ce sera l'objet du prochain chapitre). Il s'agit du diagramme à moustache et du diagramme en violon.

```
g0306 <- ggplot(foo, aes(y = trust_institut, x=1)) +
geom_boxplot(fill="Grey")

g0307 <- ggplot(foo, aes(x=1,y = trust_institut)) +
geom_violin(fill="Gold") + labs(x="density")

plot_grid(g0306, g0307, labels = c("Boxplot","Violin plot"),
  label_size = 12
)
```

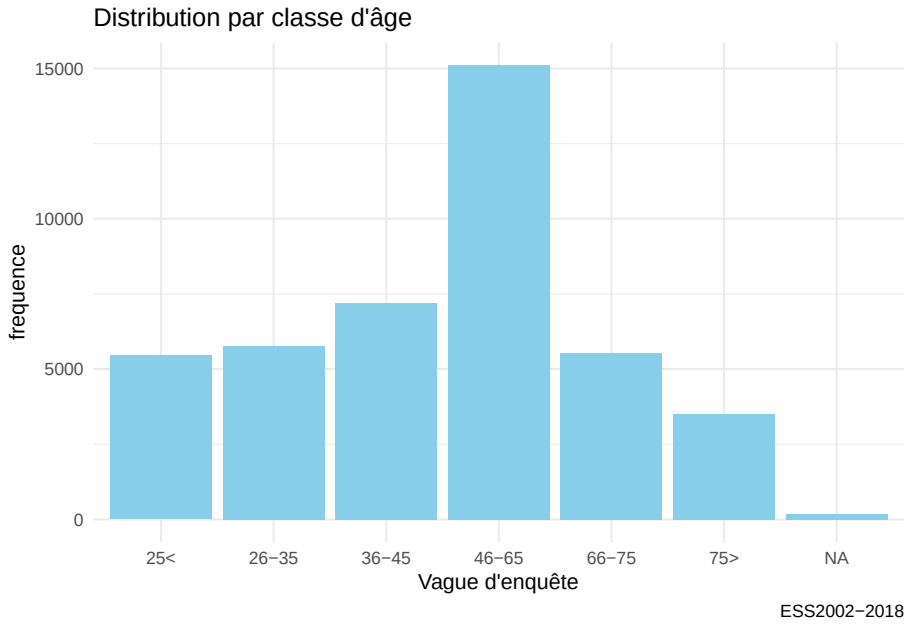


4.2.2 Quand la variable est qualitative

Quand la variable est qualitative, que ses variables sont discrètes, la manière de représenter la plus commune est le fameux camembert que les experts écartent. Un diagramme en barre représente mieux les proportions.

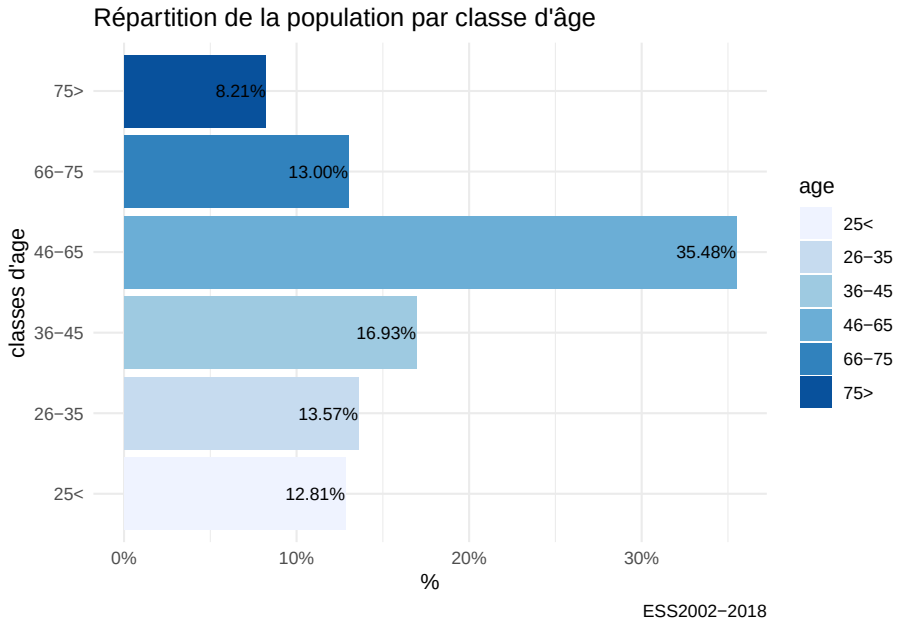
Un premier exemple pour représenter les vagues d'enquêtes

```
g08<-ggplot(df,aes(x=age))+
  geom_bar(fill="skyblue")+
  labs(title= "Distribution par classe d'âge", caption="ESS2002-2018",y= "frequence",x="Vague d'e
g08
```



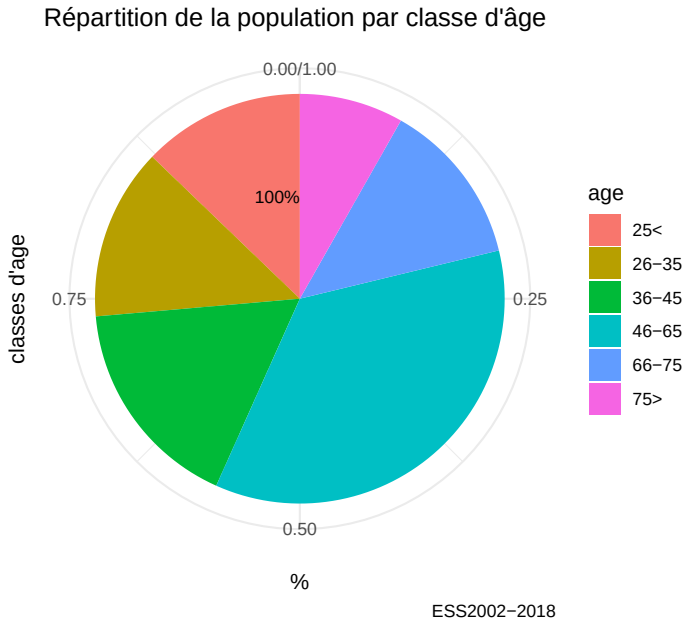
Avec quelques améliorations : contôle de la couleurs des barres, ajout des % et pivot pour une meilleure lecture.

```
foo<-df %>%
  filter(!is.na(age))

g10<-ggplot(foo,aes(x=age, y = prop.table(stat(count)),label = scales::percent(prop.tab
  geom_bar(aes(fill = age)) +
  coord_flip()+
  labs(title= "Répartition de la population par classe d'âge", caption="ESS2002-2018",y
  scale_y_continuous(labels = scales::percent)+ #contrôle de l'échelle des % et du for
  scale_fill_brewer()+
  geom_text(stat = 'count',position = position_dodge(.9),  hjust = 1, size = 3)


g10
```



si on tient au diagramme en secteur

```
foo<-df %>%filter(!is.na(age))
g10<-ggplot(foo,aes(x="", y = prop.table(stat(count)),label = scales::percent(prop.table(stat(cou
  geom_bar(aes(fill = age)) +
  labs(title= "Répartition de la population par classe d'âge", caption="ESS2002-2018",y= "%",x="c
  geom_text(stat = 'count',position = position_dodge(.9),  hjust = 1, size = 3) +
  coord_polar("y", start=0)



g10
```

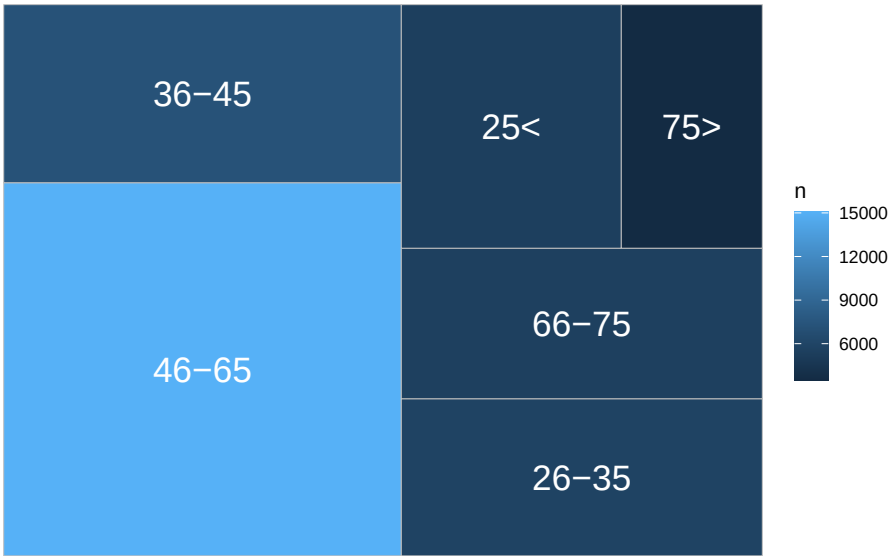


https://cran.r-project.org/web/packages/treemapify/vignettes/introduction-to-treemapify.html

si on tient au diagramme en cercle, autant opter pour un treemap avec la bibliothèque treemapifi

```
library(treemapify)
tree1<-df %>%
  mutate(n=1)%>%group_by(age) %>%
  summarize(n=sum(n)) %>%
  filter(!is.na(age))

g11 <- ggplot(tree1, aes(area = n, fill=n),label=age) +
  geom_treemap() +
  geom_treemap_text(aes(label=age),colour = "white", place = "centre",grow = FALSE)+
  labs(title= "Répartition de la population par classe d'âge", caption="ESS2002-2018",y

g11
```

Répartition de la population par classe d'âge



Chapter 5

Analyse bi variée

Comme son nom l'indique, il s'agit d'examiner la relation entre deux variables et d'étudier leur distribution conjointe. On distinguera 3 situations et on examinera pour chaune les modes de représentations graphiques ainsi que les tests associés qui permette de s'assurer que la relation apparente est effective.

a) Deux variables quantitatives : scatterplot et corélations

b) deux variable qualitatives : tableau croisé et test du chi2

c) une variable quanti et une variable quali. Compariaons de moyennes et ANOVA

d) par comparer des distribution de plusieurs groupes (variables catégorielles)

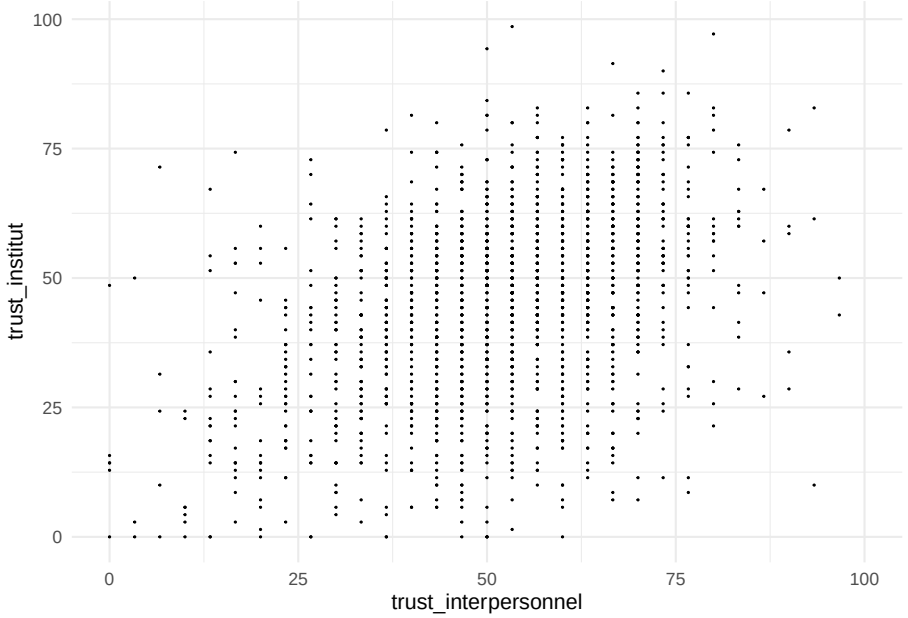
e) par comparer des moyennes d'une variable dépendante en fonction de plusieurs variables indépendantes catégorielle

f) mesurer l'association entre deux variables qualitatives

5.1 Diagrammes xy - la magie des corrélations

Venons en à analyser les relations entre deux variables quantitatives.

```
foo<-df %>%
  filter(cntry=="FR" & Year=="2018") #selection de l'echantillon

g31<- ggplot(foo, aes(x= trust_interpersonnel,y=trust_institut)) +
  geom_point( size=0.1)

g31
```



Ce graphe est peu clair, il y a trop de points qui prennent des valeurs discrètes. Une astuce est de donner une position aléatoire pour sur disperser, on fait mieux apparaitre la densité de points. On ajoute la représentation de deux courbe d'ajustement, l'une linéraire et l'autre non linéaires.

Mais en attendant en voici un calcul élémentaire.

le calcul de la variance

$$SS_{xx} = \sum (x - \bar{x})^2 = \sum x^2 - \frac{(\sum x)^2}{n}$$

le calcul de la covariance

$$SS_{xy} = \sum (x - \bar{x})(y - \bar{y}) = \sum xy - \frac{(\sum x)(\sum y)}{n}$$

et la corrélation qui est le rapport de la covariance sur la racine carrée du produit des variances de x et y.

$$r = \frac{SS_{xy}}{\sqrt{SS_{xx} SS_{yy}}}$$

La corrélation est de l'ordre d'un peu plus de 0,40 ce qui est assez élevé mais laisse une certaine indépendance des variables. Elle désignent des objets liés mais distinct. On peut tester l'hypothèse qu'en réalité cette corrélation est nulle.

Le test conduit au rejet de l'hypothèse nulle de manière très nette, compte-tenu de l'échantillon l'intervalle de confiance est compris entre 0.36 et 0.44.

```
#psych
r<-cor.test(foo$trust_interpersonnel, foo$trust_institut) #le test vient du package psych
r
```

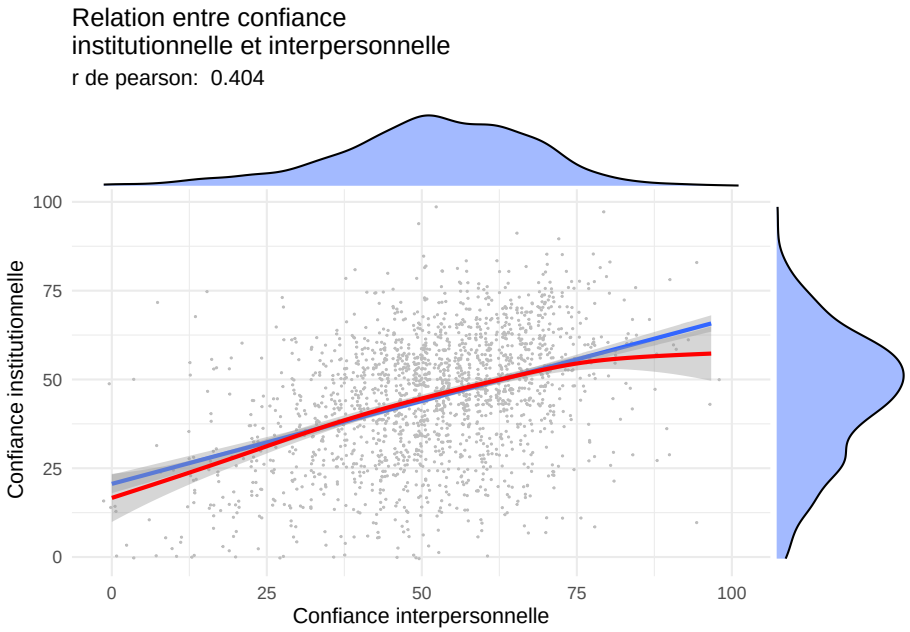
```
##
##  Pearson's product-moment correlation
##
## data:  foo$trust_interpersonnel and foo$trust_institut
## t = 18.861, df = 1821, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3651235 0.4419644
## sample estimates:
##      cor
## 0.404257
```

```
rp<-round(r$estimate,3)
rp
```

```
##   cor
## 0.404
```

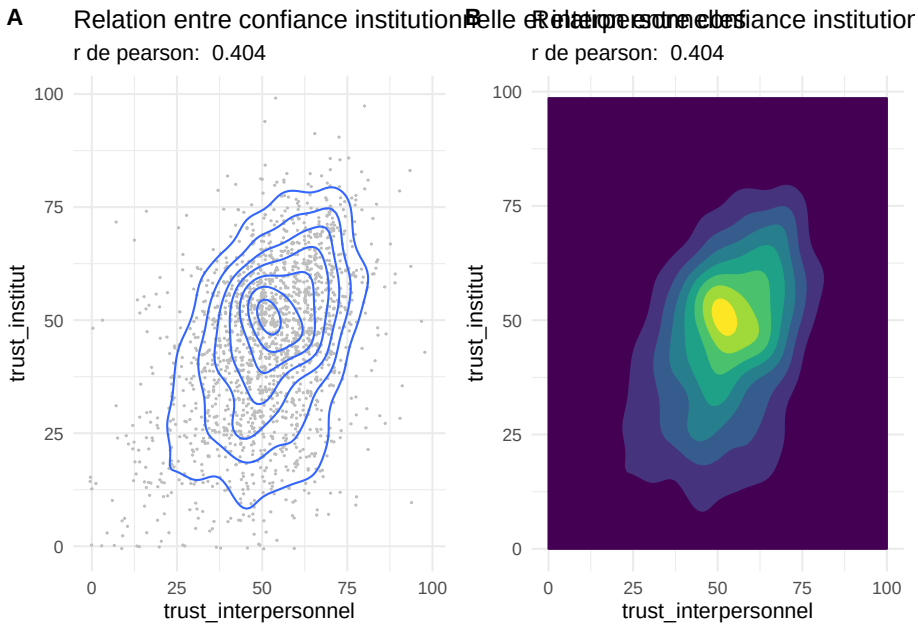
Améliorons le graphe On peut souhaiter ajouter une droite des moindre carrés (calculée pour chaque vague d'enquête pour évaluer la stabilité de la relation dans le temps). Les lignes sont parallèles, la corrélation ne change pas dans le temps, c'est une relation stable. Les deux formes de confiance vont dans le meme sens. On verra dans un autre chapitre comment calculer ces droites de corrélations.

```
library(ggExtra)
g32<-ggplot(foo, aes(x= trust_interpersonnel,y=trust_institut)) +
  geom_point(position = "jitter", size=0.1, color="grey")+
  geom_smooth(method="lm", se=TRUE) +
  geom_smooth(method="gam",color="red")     +
  labs(title = "Relation entre confiance \ninstitutionnelle et interpersonnelle",
       subtitle = paste("r de pearson: ",rp ),
       x= "Confiance interpersonnelle",
       y=" Confiance institutionnelle")

ggMarginal(g32  ,type = "density", fill = "Royalblue1", alpha=.5)
```



Une autre façon de représenter est celle de carte de densité de probabilité.

```
g32<-ggplot(foo, aes(x= trust_interpersonnel,y=trust_institut)) +
  geom_point(position = "jitter", size=0.1, color="grey")+geom_density2d()+
  labs(title = "Relation entre confiance institutionnelle et interpersonnelles", subti

g33<-ggplot(foo, aes(x= trust_interpersonnel,y=trust_institut)) +
  geom_density2d_filled(aes(fill = ..level.., color = ..level..),
    contour_var = "density")+
  labs(title = "Relation entre confiance institutionnelle et interpersonnelles", subti


plot_grid(g32, g33, labels = c('A', 'B'), label_size = 12)
```



5.2 Comparer les distributions et des moyennes

Dans notre base on a pris les données de l'Allemagne et de la France. On va comparer leur distribution. Et tant qu'à faire, puisque qu'on a deux variables, on va faire deux comparaisons : par pays et par type de confiance.

A cette fin, nous construisons un tableau de données spécifique.

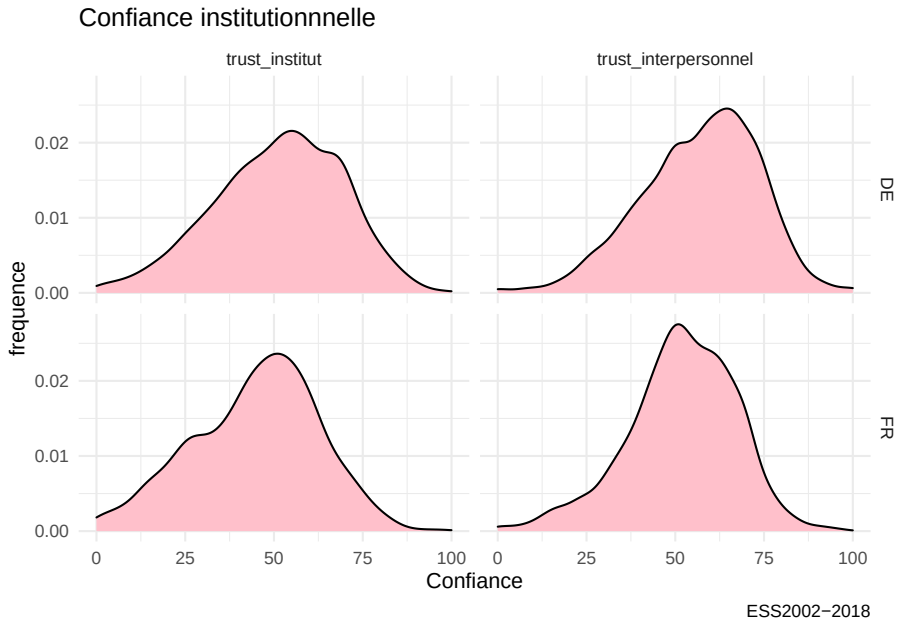
```
#on recode en facteur la variable

foo <- df %>%
  dplyr::select(cntry,trust_institut, Year,trust_interpersonnel) %>%
  filter( Year=="2018") %>%
 dplyr::select(-Year)%>%
 drop_na() %>%
  gather(variable, value, -cntry) #attention plutôt utiliser pivot_longer

head(foo)
```

```
## # A tibble: 6 x 3
##   cntry        variable       value
##   <chr+lbl>    <chr>          <dbl>
## 1 DE [Germany] trust_institut  58.6
```

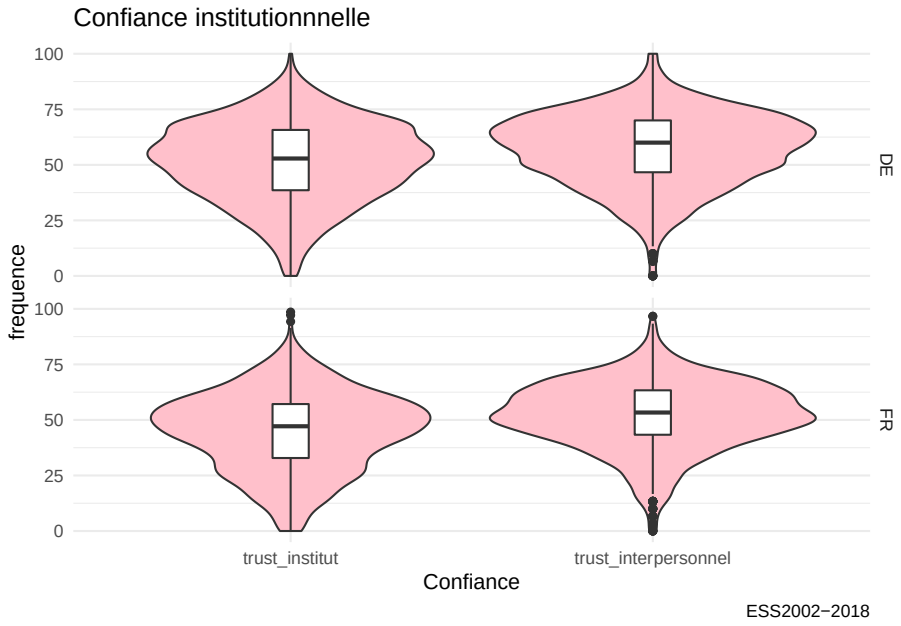
```
## 2 DE [Germany] trust_institut  65.7
## 3 DE [Germany] trust_institut  58.6
## 4 DE [Germany] trust_institut  65.7
## 5 DE [Germany] trust_institut  48.6
## 6 DE [Germany] trust_institut  37.1
```

Pour la représentation, en plus de la représentation en terme de densité, on va choisir une méthode de violon et de boxplot. On utilise une couche de "facetting" pour éclater ainsi la distribution des deux variables selon un critère de pays.

```
#on peut utiliser "facet"
g20<-ggplot(foo,aes(x=value))+ geom_density(binwidth=10, fill="pink")+ facet_grid(cntry
  labs(title= "Confiance institutionnnelle", caption="ESS2002-2018",y= "frequence",x="C
g20
```




```
g21<-ggplot(foo,aes(x=variable, y=value))+
  geom_violin( fill="pink") +
  geom_boxplot(width=0.1)+
  facet_grid(cntry~.)+
  labs(title= "Confiance institutionnnelle", caption="ESS2002-2018",y= "frequence",x="C
g21
```



5.2.1 Comparaison de moyennes

Comparer des distributions est une étape initiale nécesséaire, mais en général on sera plutôt intéresser de comparer des moyennes. Par exemple, on souhaiterais savoir si les degrés de confiances institutionnnelle et interpersonnelles varient en France selon les situations de revenu.

Calculons d'abord ces moyennes avec la fonction group_by et summarise.

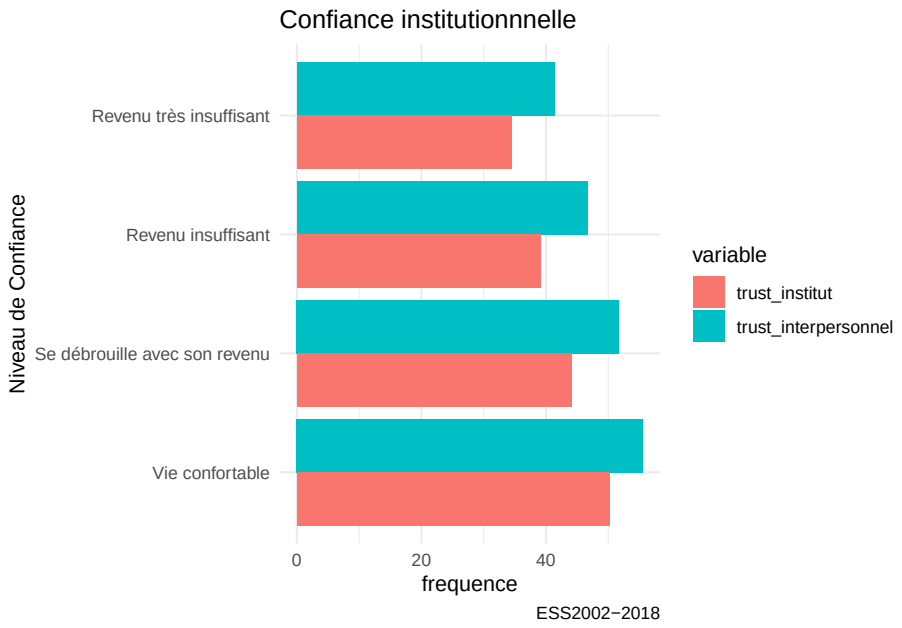
```
df_wave<-df %>% filter(cntry=="FR" & Year=="2018") %>%
  group_by(revenu) %>%
  summarise(trust_interpersonnel=mean(trust_interpersonnel, na.rm=TRUE),
            trust_institut =mean(trust_institut, na.rm=TRUE)) %>%
  filter(!is.na(revenu)) %>%                                    #filtrer les valeurs ma
  gather(variable, value, -revenu)                              #fichier long ( pivot l
head(df_wave)
```

```
## # A tibble: 6 x 3
##   revenu                          variable             value
##   <fct>                           <chr>                <dbl>
## 1 Vie confortable                 trust_interpersonnel  55.6
## 2 Se débrouille avec son revenu   trust_interpersonnel  51.7
## 3 Revenu insuffisant              trust_interpersonnel  46.7
## 4 Revenu très insuffisant         trust_interpersonnel  41.4
```

```
## 5 Vie confortable               trust_institut        50.2
## 6 Se débrouille avec son revenu trust_institut        44.1
```

Représentons ces moyennes graphiquement avec un geom_bar.

```
g06a<-ggplot(df_wave,aes(x=revenu,y=value, group=variable))+
  geom_bar(stat="identity",aes(fill=variable), position =position_dodge())+
  labs(title= "Confiance institutionnnelle", caption="ESS2002-2018",y= "frequence",x="
  coord_flip()

g06a
```



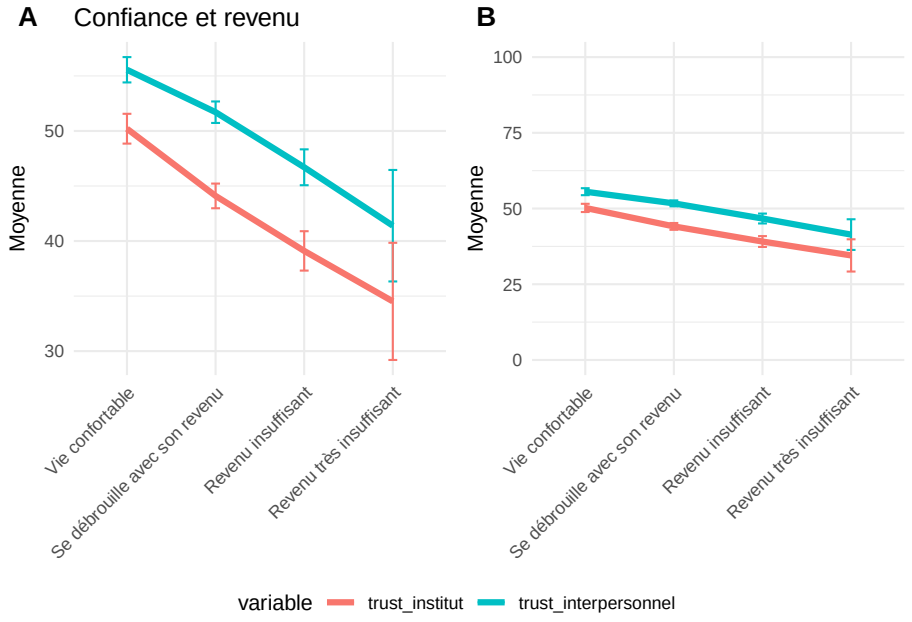
On a une solution mais pas la meilleure, on perd l'idée de variance et ce serait bien d'ajouter des barres d'intervalle de confiances , un diagramme en lignes serait plus élégant. On en profite pour corriger l'aspect des labels peu lisibles en les inclinants, et à choisir une échelle qui omettent les valeur supérieur à 70 et inférieure à 30 pour donner une vision plus respectueuses de la totalité de l'échelle qui va de 0 à 100.

Au passage on emploie à nouveau cowplot pour combiner les graphes, et ici plus précisément partager la légende des deux graphiques.

On observera que si le niveau de confiance diminue avec le revenu, la confiance interpersonnelle est plus forte, et de manière parallèle, à la confiance institutionnelle. On remarquera enfin que c'est pour les revenu les plus faibles que l'estimation est la plus imprécise ou la variance la plus grande.

```
df_wave2<-df %>%
  filter(cntry=="FR" & Year=="2018")%>%
  group_by(revenu) %>%
  mutate(n=1) %>%
  summarise(trust_interpersonnel_se=sd(trust_interpersonnel, na.rm=TRUE), #on calcule l'écartype
            trust_institut_se =sd(trust_institut, na.rm=TRUE),
            n=sum(n),
            trust_interpersonnel_se= 2*trust_interpersonnel_se/sqrt(n), # on calcule l'erreur typ
            trust_institut_se=2*trust_institut_se/sqrt(n)
            ) %>% dplyr::select(-n) %>%
  filter(!is.na(revenu)) %>%
  gather(variable, value, -revenu) %>% #on passe en format long
  dplyr::select(-revenu,-variable)%>%
  rename(se=value)

df_wave3<-cbind(df_wave,df_wave2) #on concatène les moyennes et les erreurs types

#on peut enfin produire le graphique

g06a<-ggplot(df_wave3,aes(x=revenu,y=value, group=variable))+
  geom_line(stat="identity",aes(color=variable), size=1.5)+
  geom_errorbar(aes(ymin=value-se, ymax=value+se, color=variable), width=.2,position=position_doc
  labs(title= "Confiance et revenu",y= "Moyenne",x=NULL)+
  theme(axis.text.x = element_text( angle=45, hjust =1)) #on controle l'angle et la position hori


g06b<-ggplot(df_wave3,aes(x=revenu,y=value, group=variable))+
  geom_line(stat="identity",aes(color=variable), size=1.5)+
  geom_errorbar(aes(ymin=value-se, ymax=value+se, color=variable), width=.2,position=position_doc
  ylim(0,100)+
  labs(title= "",y= "Moyenne",x=NULL)+
  theme(axis.text.x = element_text( angle=45, hjust =1)) #on controle l'angle et la position hori

prow <- plot_grid(
  g06a + theme(legend.position="none"),
  g06b + theme(legend.position="none"),
  align = 'vh',
  labels = c("A", "B", "C"),
  hjust = -1,
  nrow = 1
)
# extract a legend that is laid out horizontally
legend_b <- get_legend(
  g06a +
    guides(color = guide_legend(nrow = 1)) +
```

```
    theme(legend.position = "bottom")
)

# add the legend underneath the row we made earlier. Give it 10%
# of the height of one plot (via rel_heights).
plot_grid(prow, legend_b, ncol = 1, rel_heights = c(1, .1))
```



La visualisation est utile, encore faut-il qu'on soit bien certain que les variations ne soit pas le produit du hasard, des fluctuations d'échantillonnage. Si en moyenne la perception du pouvoir d'achat est associée à des moyennes de confiance décroissantes, les différences observées sont-elle significatives? Dans les représentations précédentes c'est le choix de l'échelle qui oriente l'analyse.

On a un besoin d'un test plus objectif. Celui est le très classique test d'analyse de variance (ANOVA).

Celui-çi est le test d'analyse de variance qui consiste à comparer la variance à l'intérieur des groupes (intra), et la variance entre les moyennes des groupes (inter ou between).

On note qu'ici on introduit la méthode flextable pour présenter des tableaux au formats scientifique. L'astuce ici est d'utiliser aussi xtable.

```
foo<-df %>%
  filter(cntry=="FR" & Year=="2018") %>% drop_na() #selection des données
```

```
fit<-lm(trust_institut~revenu, foo) #calcul du modèle linéaire

anova(fit) #test d'analyse de variance
```

```
## Analysis of Variance Table
##
## Response: trust_institut
##             Df Sum Sq Mean Sq F value    Pr(>F)
## revenu       3  27651  9217.1  32.052 < 2.2e-16 ***
## Residuals 1686 484846   287.6
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
library(xtable) #xtable transforme en table certains type d'objet dont les résultats de l'anova
ft <- xtable_to_flextable(xtable(anova(fit)), hline.after = c(0,2)) #la fonction permet d'exploi
ft
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
revenu	3	27,651.4	9,217.1	32.1	0.0
Residuals	1,686	484,845.7	287.6		

5.2.2 Deux variables qualitatives

L'étude de la relation éventuelle entre deux variables qualitative s'apprécie traditionnellement par une méthode de tableau croisé.

5.2.2.1 Tableau croisé

Pour calculer le tableau croisé on utilise la fonction très simple `table` et la fonction `prop.table`

```
t<-table(foo$revenu,foo$habitat)
t
```

```
##
##                                Big city Suburbs Town Village Countryside
##   Vie confortable                   118      82  161     142          31
##   Se débrouille avec son revenu     120     109  275     227          58
##   Revenu insuffisant                 48      38  129      88          22
##   Revenu très insuffisant             9       5   18      10           0
```

```
prop.table(t,2)
```

```
##
##                                   Big city    Suburbs       Town    Village
##   Vie confortable                0.40000000 0.35042735 0.27615780 0.30406852
##   Se débrouille avec son revenu  0.40677966 0.46581197 0.47169811 0.48608137
##   Revenu insuffisant             0.16271186 0.16239316 0.22126930 0.18843683
##   Revenu très insuffisant        0.03050847 0.02136752 0.03087479 0.02141328
##
##                                  Countryside
##   Vie confortable                 0.27927928
##   Se débrouille avec son revenu   0.52252252
##   Revenu insuffisant              0.19819820
##   Revenu très insuffisant         0.00000000
```

Mais ce n'est pas esthétique, avec la fonction `proc_freq` de flextable on obtient une meilleure présentation. Elle nous donne en peu de mots les effectif par cellule, les pourcentages en lignes, et en colonnes.

```
ft1<- proc_freq(foo, "revenu", "habitat", include.table_percent = FALSE,
                include.row_percent = FALSE, include.column_total = FALSE,
  include.column_percent = TRUE)
ft1
```

revenu	label	Big city	Suburbs
Vie confortable	Frequency	118	82
	Col Pct	40%	35.04%
Se débrouille avec son revenu	Frequency	120	109
	Col Pct	40.68%	46.58%
Revenu insuffisant	Frequency	48	38
	Col Pct	16.27%	16.24%
Revenu très insuffisant	Frequency	9	5
	Col Pct	3.05%	2.14%

```
ft2<- proc_freq(foo, "revenu", "habitat", include.table_percent = FALSE,
                include.row_percent = TRUE,
  include.column_percent = FALSE)
ft2
```

hab

revenu	label	Big city	Suburbs	Town
Vie confortable	Frequency	118	82	161
	Row Pct	22.1%	15.36%	30.15%
Se débrouille avec son revenu	Frequency	120	109	275
	Row Pct	15.21%	13.81%	34.85%
Revenu insuffisant	Frequency	48	38	129
	Row Pct	14.77%	11.69%	39.69%
Revenu très insuffisant	Frequency	9	5	18
	Row Pct	21.43%	11.9%	42.86%
Total	Frequency	295	234	583

5.2.2.2 le valeureux chi²

Le test du chi2 s'appuie sur une idée très simple qui de fait est un théorème : Si deux variables X et Y sont indépendantes, la fréquence de leur combinaison est le produit des fréquences marginales.

On peut donc sur cette base, calculer l'effectif attendu (expected frequency) puis le comparer à ce qu'on a observé pour chacune des cellules du tableau. On somme enfin ces écarts.

$$\chi^2 = \sum \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

Naturellement , une même valeur de cette quantité pour un petit tableau(2x2) n'a pas la même signification que si le tableau est grand(par ex 20x 10). On l'appréciera donc en fonction des degrés de liberté (n-1 x m-1).

Le test proprement dit consiste à examiner quelles sont les chances qu'on obtienne la valeur du chi2 calculé, pour un nombre de degré de liberté donné. Si cette probabilité est faible on rejetera l'hypothèse d'indépendance des deux variables.

Avec r la fonction chsq.test nous simplifie

```
chi2<-chisq.test(t)
chi2
```

```
##
```

```
##  Pearson's Chi-squared test
##
## data:  t
## X-squared = 23.853, df = 12, p-value = 0.0213
```

L'objet chi2 est une liste

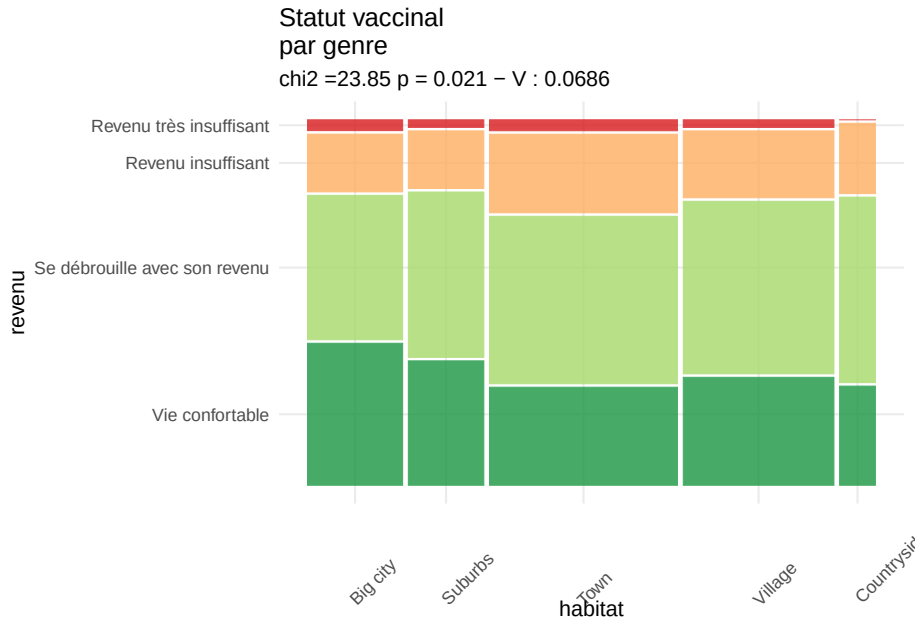
```
# On isole les éléments qui nous intéresse

#library()
chi<-round(chi2$statistic,2)
p<-round(chi2$p.value,3)
V<-cramerV(t, digit=3)
```

5.2.2.3 diagramme en mosaique

Le diagramme en mosaique détermine la largeur des barres en fonction de l'effectif de la variable en abcisse et leur hauteur en fonction de la variable en ordonnée. Les couleurs permettent de mieux comparer.

On s'aperçoit ici que les plus à l'aise avec leur revenu sont proportionnellement plus nombreux dans les grandes villes, et que ceux qui se débrouille sont plus fréquents dans les campagnes.

```
library(ggmosaic)
g1 <- ggplot(data = foo) +
  geom_mosaic(aes(x=product( revenu ,habitat), fill = revenu))+
  theme(axis.text.x = element_text(angle = 45, hjust = -0.1, vjust = -0.2))+
  theme(legend.position = "none")+
  labs(title="Statut vaccinal \npar genre",
       subtitle=paste0("chi2 =",chi, " p = ", p, " - V : ", V))+
  scale_fill_brewer(palette = "RdYlGn", direction = -1)

g1
```



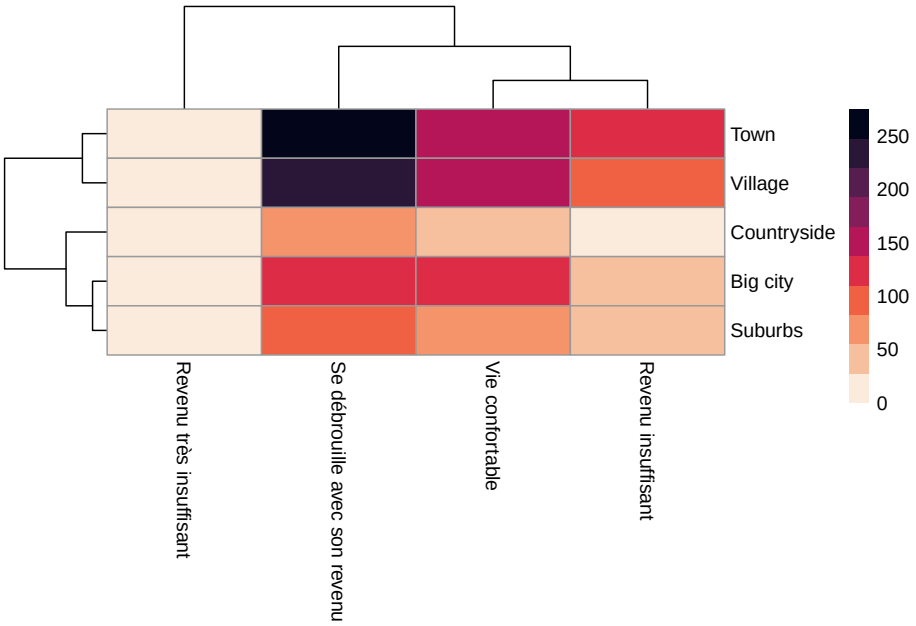
5.2.2.4 les chi2s partiel et des cartes de chaleur.

Une carte de chaleur représente une grandeur par un gradient de couleur pour chaque cellule définie par des variable x et y.

Faisons un premier essai pour représenter les effectifs, plutôt qu'avoir un tableau de nombres on va obtenir un tableau de couleurs.

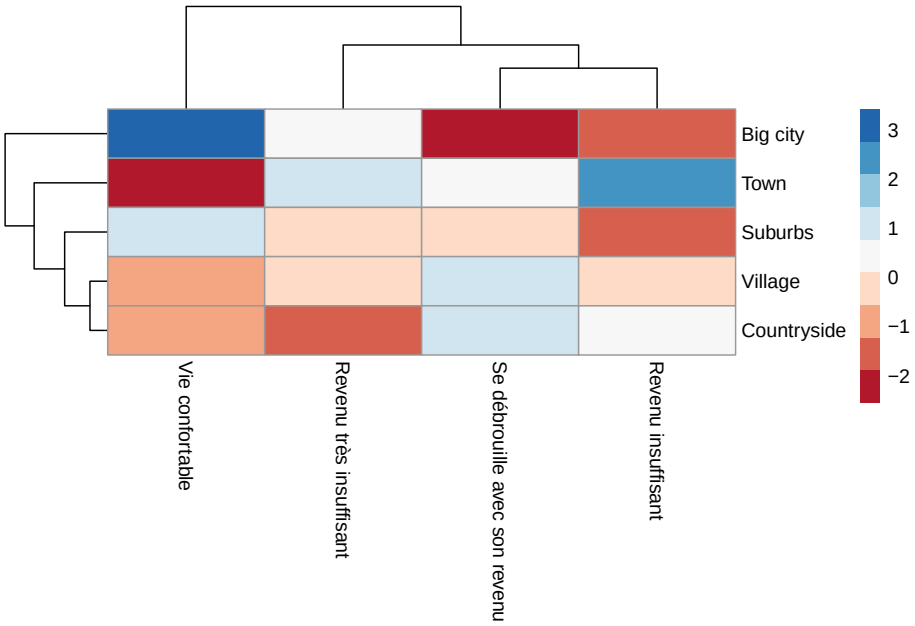
L'arbre qui apparait en ligne et en colonne correspond au résultat d'une classification hiérarchique que nous développons dans le chapitre X.

```
library(pheatmap)
library(viridis)

table2<-as.data.frame(t) %>%
  pivot_wider(names_from = Var1, values_from = Freq) %>%
  column_to_rownames( var = "Var2")
pheatmap(table2 , color = rocket(10,direction =-1))
```



On utilise la même technique mais en représenant une grandeur différentes : les tests du chi2 partiels, pour apprécier les sous ou les sur-représentation.

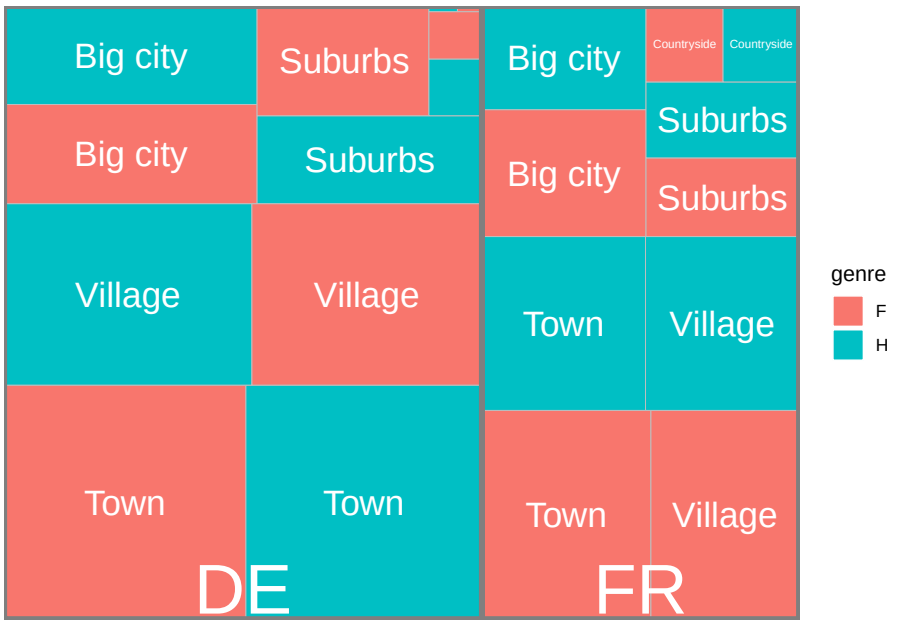
```
library(RColorBrewer)
chi2df<- as.data.frame(chi2$stdres)

table2<-chi2df %>%
  pivot_wider(names_from = Var1, values_from = Freq) %>%
  column_to_rownames( var = "Var2")
pheatmap(table2 , color = brewer.pal(n = 9, name = "RdBu"))
```



5.2.2.5 Les treemaps, c'est merveilleux

D'autre graphiques et des emboitements

```
library(treemapify)
tree1<-df %>% mutate(n=1)%>%group_by(cntry,genre,habitat) %>% summarize(n=sum(n),mean=mean(trust_

g10 <- ggplot(tree1, aes(area = n, fill=genre,subgroup=cntry)) +
  geom_treemap() +
  geom_treemap_text(aes(label=habitat),colour = "white", place = "centre",grow = FALSE)+
      geom_treemap_subgroup_text(color="white",grow = FALSE)+
  geom_treemap_subgroup_border()
g10
```


Big city
Suburbs
Big city
Suburbs
Village
Village
Town
Town
DE
Big city
Countryside
Countryside
Suburbs
Big city
Suburbs
Town
Village
Town
Village
FR
genre
F
H

Chapter 6

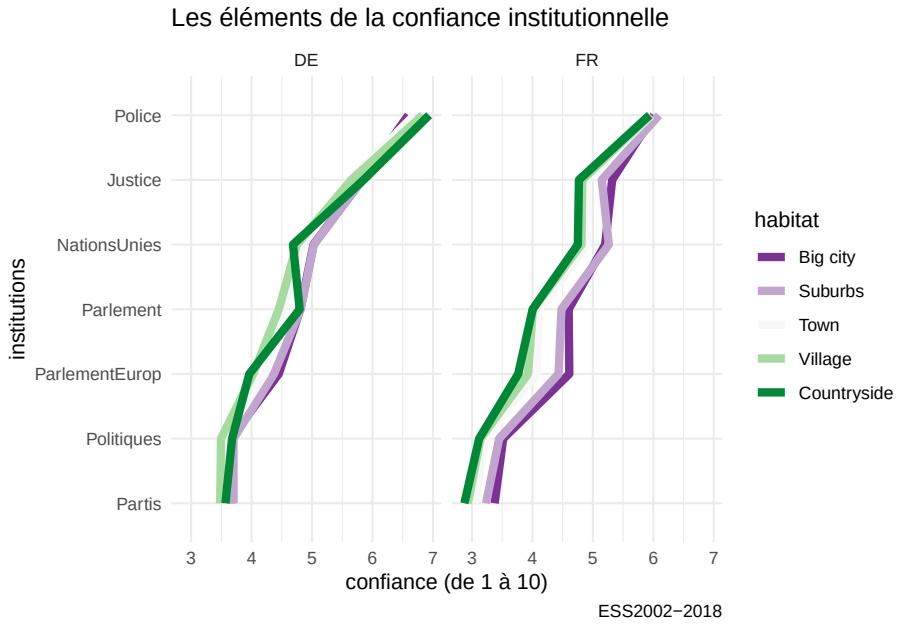
Analyse graphique multivariée

Dans ce chapitre, on généralise à des ensembles de variables.

6.1 La confiance institutionnelle, en détail

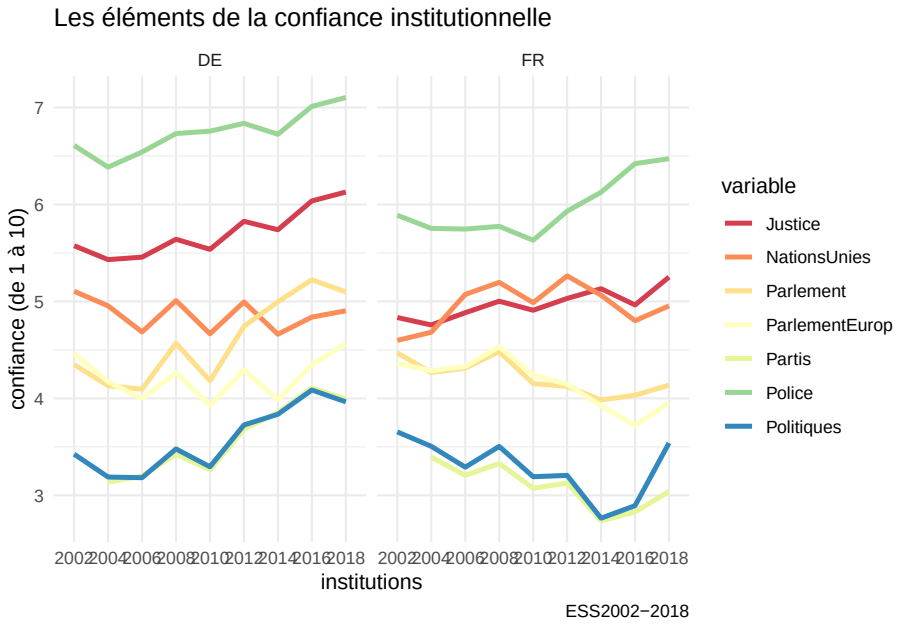
On veut reprénter 6 variables, correspondant à 5 types d'habitats et 2 pays.

```
df<-readRDS("./data/dfTrust.rds)")


rad<-df %>%
  group_by (habitat,cntry) %>%
  summarize(Partis=mean(Partis, na.rm=TRUE),
  Parlement=mean(Parlement, na.rm=TRUE),
  Politiques=mean(Politiques, na.rm=TRUE),
  Police=mean(Police, na.rm=TRUE),
  Justice=mean(Justice, na.rm=TRUE),
  NationsUnies=mean(NationsUnies, na.rm=TRUE),
  ParlementEurop=mean(ParlementEurop, na.rm=TRUE)) %>%
  filter(!is.na(habitat)) %>%
  gather(variable, value, -habitat, -cntry)

ggplot(rad, aes(x=reorder(variable, value),y=value, group=habitat))+
  geom_line(aes(color=habitat), size=2)+
  facet_grid(.~cntry) +coord_flip()+
  scale_color_brewer(type="div",palette=3)+labs(title= "Les éléments de la confiance institutionn
```



Une autre variante qui donne l'évolution de l'évolution de les éléments de la confiance institutionnelle

```
rad<-df %>%
  group_by (Year,cntry) %>%
  summarize(Partis=mean(Partis, na.rm=TRUE),
  Parlement=mean(Parlement, na.rm=TRUE),
  Politiques=mean(Politiques, na.rm=TRUE),
  Police=mean(Police, na.rm=TRUE),
  Justice=mean(Justice, na.rm=TRUE),
  NationsUnies=mean(NationsUnies, na.rm=TRUE),
  ParlementEurop=mean(ParlementEurop, na.rm=TRUE)) %>%
  gather(variable, value, -Year, -cntry)

ggplot(rad, aes(x=Year,y=value, group=variable))+
  geom_line(aes(color=variable), size=1.2)+
  facet_wrap(.~cntry, nrow=1) +
  scale_color_brewer(palette="Spectral")+labs(title= "Les éléments de la confiance inst
```

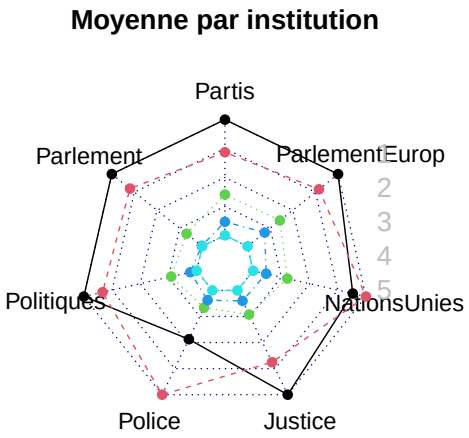


La différence entre les deux pays est claire, la rupture est accusée plus fortement en France qu'en Allemagne. L'explication n'est sans doute pas culturelle mais démographique, un coup d'oeil à la carte des densité permet de comprendre mieux : https://www.populationdata.net/cartes/allemagne-france-densite-de-population-2011/.

On pourra tenté un graphe en radar. Mais il n'est pas si convaincant.

```
library(fmsb)

rad<-df %>% filter(cntry=="FR") %>%
  group_by (habitat) %>%
  summarize(Partis=mean(Partis, na.rm=TRUE),
  Parlement=mean(Parlement, na.rm=TRUE),
  Politiques=mean(Politiques, na.rm=TRUE),
  Police=mean(Police, na.rm=TRUE),
  Justice=mean(Justice, na.rm=TRUE),
  NationsUnies=mean(NationsUnies, na.rm=TRUE),
  ParlementEurop=mean(ParlementEurop, na.rm=TRUE)) %>%
  filter(!is.na(habitat)) %>%
  dplyr::select(-habitat)

#on doit indiquer les valeurs minimale et maximale - la fonction rep permet de repeter (ici 7 fo
data <- rbind(rep(7,7) , rep(3,7) , rad)
#l'autre method c'est ce choisir maxmin=FALSE
```

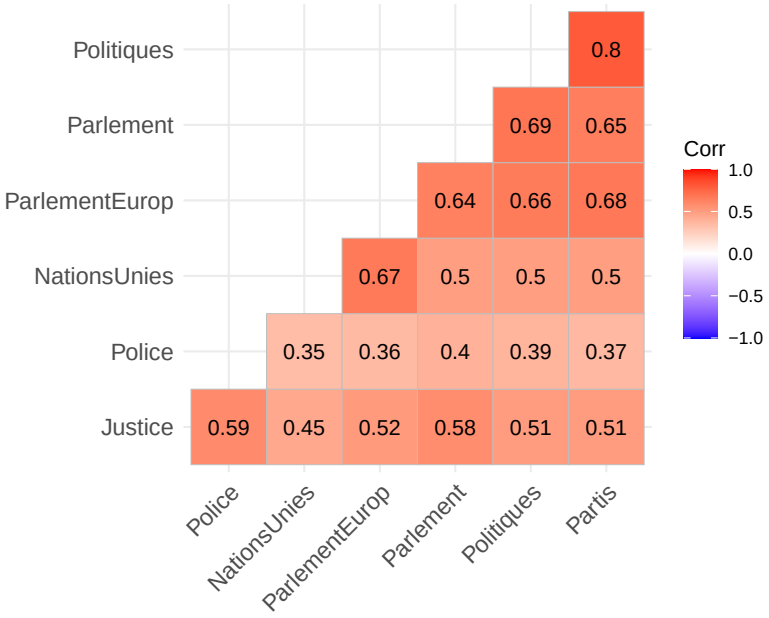
```
#rownames(rad) <- c("big city", "suburbs" ,"town","village", "countryside")
radarchart(rad, axistype=0, seg=4, title="Moyenne par institution", maxmin=FALSE)
legend(x=0.7, y=1, legend = rownames(rad), bty = "n",text.col = "grey", cex=1.2, pt.cex
```



6.2 Table de corrélation

Comparer les moyennes est une chose, on souhaiter en plus savoir quelle structure de corrélation les caractérisent. Rien de plus simple

```
library(ggcorrplot)
df<-readRDS("./data/dfTrust.rds)")%>%filter(Year==2018)

foo<-df %>% dplyr::select(NationsUnies,ParlementEurop, Parlement, Justice, Police, Poli
  drop_na()
r<-cor(foo)

ggcorrplot(r, hc.order = TRUE, type = "lower",
   lab = TRUE)
```




```
g<-paste0("./plot/g1",".jpg")
ggsave(g,plot=last_plot(), width = 27, height = 19, units = "cm")
```

6.3 Un cas plus complexe : présidentielle2020

Nsppolls cumulent les sondages publiés des grands instituts. On utilise ces données , ainsi qu'une boucle, pour explorer différents paramètre d'un modèle de lissage.

Le but : mieux percevoir les tendance par une sorte de méta-analyse des différents sondages :

6.4 une boucle pour produire de multiple graphe en variant un paramètre

```
library(lubridate)
alph<-.5

for (alph in seq(from=0, to= 1, by=.05)){
df_pol <- read_delim("https://raw.githubusercontent.com/nsppolls/nsppolls/master/presidentielle.c
```

```
                      delim = ",", escape_double = FALSE, trim_ws = TRUE)%>%
  filter(tour=="Premier tour") %>%filter(candidat=="Eric Zemmour"|
                                           candidat== "Marine Le Pen"|
                                           candidat== "Emmanuel Macron"|
                                           candidat== "Jean-Luc Mélenchon"|
                                           candidat== "Yannick Jadot"|
                                           candidat== "Valérie Pécresse"|
                                           candidat=="Fabien Roussel"|
                                           candidat=="Anne Hidalgo") %>%
  filter(fin_enquete>ymd("2022-01-09")) # on commence en septembre , octobre est-il me


table(df_pol$candidat)
SensiP1<-c("pink", "orange", "gray20", "red","firebrick", "Royalblue", " skyblue", "Cha

ggplot(df_pol, aes(y=intentions, x=fin_enquete))+
  geom_point(aes(color=candidat), size=.5, alpha=1-alph)+
  geom_smooth(span = alph, aes(col=candidat,fill=candidat), alpha=0.2)+
  scale_color_manual(values=SensiP1)+
  scale_fill_manual(values=SensiP1)+
  labs(title= "Evolution des intentions de vote #présidentielle2022 1er tour",
       subtitle =paste("Lissage méthode loess. alpha=",alph, " - ci=95%"),
       caption = "data @nsppolls viz @benavent",
       x=NULL)+theme_minimal()+scale_x_date(date_breaks = "1 month", date_minor_breaks
             date_labels = "%B")

sondage_nsppolls<-paste0("./nsppolls/sondage_nsppolls", alph*20, ".jpg")
ggsave(sondage_nsppolls,plot=last_plot(), width = 27, height = 19, units = "cm")

}


n<-df_pol%>%
  mutate(n=1)%>%
  group_by(id)%>%summarise(n=sum(n))
#nombre de sondage
n<-nrow(n)
```

Pour créer le gif on emplie magick. On a pris soin de sauvegarder les graphes dans un répertoire propre, ça facilite la lecture en boucle et la fabrication du gif.

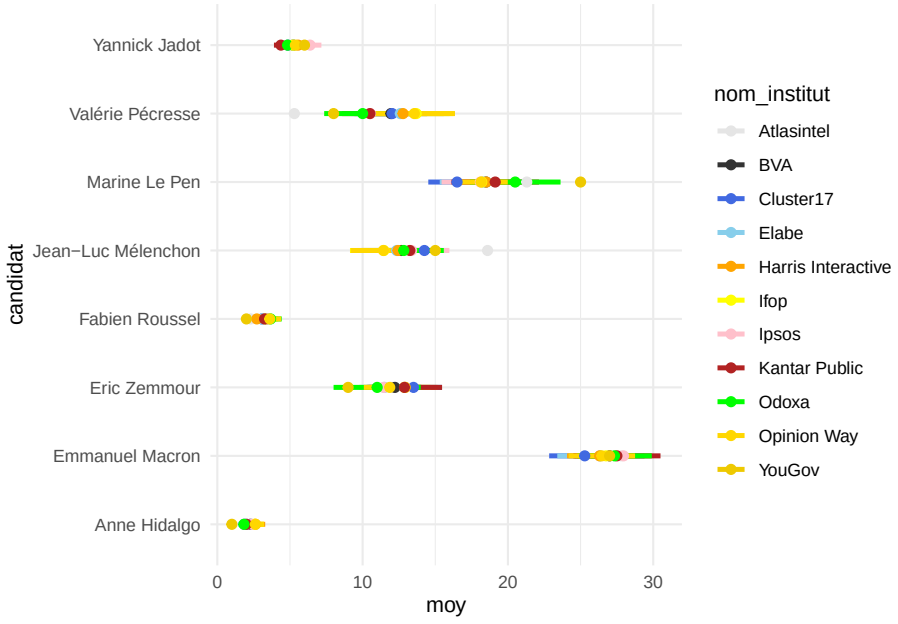
```
library(magick)

#gif
```

```
#on constitue une liste des noms des fichier *.jpg que l'on veut associer
frames <- paste0("./nsppolls/","sondage_nsppolls", 0:20,".jpg")

#on lit et on stoke dans m les images
m <- image_read(frames)

#on fabrique et on sauvergarde le gif
m <- image_animate(m, fps=1)
image_write(m, "./plot/sondages_lissage.gif")
```

6.4.1 effet sondeur

pour anticiper sur le chapitre suivant

```
foo<-df_pol%>%
  dplyr::select(candidat, intentions, fin_enquete, echantillon,nom_institut)%>%
  group_by(nom_institut, candidat)%>%
  summarise(moy=mean(intentions, na.rm=TRUE),
            std=sd(intentions, na.rm=TRUE))


SensiP2<-c("gray90","gray20", "Royalblue", "skyblue", "orange", "yellow", "pink", "firebrick", "g


g<-ggplot(foo,aes(x=candidat,y=moy))+
  geom_segment(aes(x = candidat,
                   y = -std+moy,
                   xend = candidat,
                   yend = std+moy,
                   color = nom_institut), size=1.2)+
    geom_point(aes(color=nom_institut), size=2)+
  scale_color_manual(values = SensiP2)+
  theme_minimal()+
  coord_flip()
g
```



6.5 Modéliser le biais du sondeur

http://www.stat.yale.edu/Courses/1997-98/101/anovareg.htm

```
df_pol$tps<-2
df_pol$tps[df_pol$fin_enquete < ymd("2022-01-31")]<-1

df_pol$tps[df_pol$fin_enquete > ymd("2022-03-01")]<-3

df_pol$tps<- as.factor(df_pol$tps)

fit1<- lm(intentions~candidat*tps,data=df_pol)
anova(fit1)
```

```
## Analysis of Variance Table
##
## Response: intentions
##                Df Sum Sq Mean Sq    F value    Pr(>F)
## candidat        7 122735 17533.6 10705.9435 < 2.2e-16 ***
## tps             2     25    12.7     7.7658 0.0004363 ***
## candidat:tps   14   4534   323.9   197.7554 < 2.2e-16 ***
## Residuals    2096   3433     1.6
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
fit2<- lm(intentions~candidat*tps+candidat*nom_institut,data=df_pol)
anova(fit2)
```

```
## Analysis of Variance Table
##
## Response: intentions
##                          Df Sum Sq Mean Sq    F value    Pr(>F)
## candidat                  7 122735 17533.6 12742.2074 < 2.2e-16 ***
## tps                       2     25    12.7     9.2428  0.000101 ***
## nom_institut             10     25     2.5     1.8283  0.051096 .
## candidat:tps             14   4534   323.9   235.3684 < 2.2e-16 ***
## candidat:nom_institut    70    633     9.0     6.5768 < 2.2e-16 ***
## Residuals              2016   2774     1.4
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(fit1,fit2)
```

```
## Analysis of Variance Table
##
## Model 1: intentions ~ candidat * tps
## Model 2: intentions ~ candidat * tps + candidat * nom_institut
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1   2096 3432.7
## 2   2016 2774.1 80    658.65 5.9832 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(fit1)
```

```
##
## Call:
## lm(formula = intentions ~ candidat * tps, data = df_pol)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.2585 -0.6350 -0.0278  0.6021  5.6493
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      3.1765     0.1792  17.726  < 2e-16 ***
## candidatEmmanuel Macron         21.5392     0.2534  84.992  < 2e-16 ***
```

```
## candidatEric Zemmour              9.7549     0.2534  38.492  < 2e-16 ***
## candidatFabien Roussel           -0.8039     0.2534  -3.172 0.001535 **
## candidatJean-Luc Mélenchon        6.4118     0.2534  25.300  < 2e-16 ***
## candidatMarine Le Pen            13.7353     0.2534  54.198  < 2e-16 ***
## candidatValérie Pécresse         13.2255     0.2534  52.187  < 2e-16 ***
## candidatYannick Jadot             2.7255     0.2534  10.755  < 2e-16 ***
## tps2                             -0.6765     0.2342  -2.888 0.003915 **
## tps3                             -0.9765     0.2089  -4.674 3.14e-06 ***
## candidatEmmanuel Macron:tps2      0.6844     0.3312   2.066 0.038934 *
## candidatEric Zemmour:tps2         2.0645     0.3312   6.233 5.52e-10 ***
## candidatFabien Roussel:tps2       2.1511     0.3312   6.494 1.04e-10 ***
## candidatJean-Luc Mélenchon:tps2   1.6438     0.3312   4.963 7.52e-07 ***
## candidatMarine Le Pen:tps2        0.5842     0.3312   1.764 0.077955 .
## candidatValérie Pécresse:tps2    -0.9616     0.3312  -2.903 0.003734 **
## candidatYannick Jadot:tps2       -0.1630     0.3312  -0.492 0.622725
## candidatEmmanuel Macron:tps3      4.6819     0.2955  15.847  < 2e-16 ***
## candidatEric Zemmour:tps3        -1.0570     0.2955  -3.578 0.000355 ***
## candidatFabien Roussel:tps3       2.0919     0.2955   7.080 1.95e-12 ***
## candidatJean-Luc Mélenchon:tps3   5.1319     0.2955  17.370  < 2e-16 ***
## candidatMarine Le Pen:tps3        3.4154     0.2955  11.560  < 2e-16 ***
## candidatValérie Pécresse:tps3    -4.8670     0.2955 -16.473  < 2e-16 ***
## candidatYannick Jadot:tps3        0.4724     0.2955   1.599 0.109995
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.28 on 2096 degrees of freedom
## Multiple R-squared:  0.9737, Adjusted R-squared:  0.9735
## F-statistic:  3379 on 23 and 2096 DF,  p-value: < 2.2e-16
```

```
summary(fit2)
```

```
##
## Call:
## lm(formula = intentions ~ candidat * tps + candidat * nom_institut,
##     data = df_pol)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.7981 -0.5456 -0.0272  0.5924  5.2019
##
## Coefficients:
##                                                        Estimate Std. Error
## (Intercept)                                             3.39409    1.18971
## candidatEmmanuel Macron                                20.60082    1.68251
## candidatEric Zemmour                                   10.59844    1.68251
```

```
## candidatFabien Roussel                                  -2.19508    1.68251
## candidatJean-Luc Mélenchon                              11.04593    1.68251
## candidatMarine Le Pen                                   15.30968    1.68251
## candidatValérie Pécresse                                 7.55835    1.68251
## candidatYannick Jadot                                    2.30826    1.68251
## tps2                                                    -0.68106    0.21565
## tps3                                                    -0.99409    0.19847
## nom_institutBVA                                         -0.30224    1.23753
## nom_institutCluster17                                   -0.84531    1.21657
## nom_institutElabe                                       -0.71509    1.21215
## nom_institutHarris Interactive                          -0.18347    1.21291
## nom_institutIfop                                        -0.18249    1.18474
## nom_institutIpsos                                       -0.08904    1.19038
## nom_institutKantar Public                               -0.47826    1.31223
## nom_institutOdoxa                                       -0.67101    1.35576
## nom_institutOpinion Way                                 -0.04925    1.18126
## nom_institutYouGov                                      -1.40000    1.65893
## candidatEmmanuel Macron:tps2                             0.73921    0.30498
## candidatEric Zemmour:tps2                                2.12294    0.30498
## candidatFabien Roussel:tps2                              2.12788    0.30498
## candidatJean-Luc Mélenchon:tps2                          1.71555    0.30498
## candidatMarine Le Pen:tps2                               0.60025    0.30498
## candidatValérie Pécresse:tps2                           -0.97779    0.30498
## candidatYannick Jadot:tps2                              -0.15932    0.30498
## candidatEmmanuel Macron:tps3                             4.79918    0.28068
## candidatEric Zemmour:tps3                               -0.99844    0.28068
## candidatFabien Roussel:tps3                              2.09508    0.28068
## candidatJean-Luc Mélenchon:tps3                          5.15407    0.28068
## candidatMarine Le Pen:tps3                               3.59032    0.28068
## candidatValérie Pécresse:tps3                           -4.65835    0.28068
## candidatYannick Jadot:tps3                               0.29174    0.28068
## candidatEmmanuel Macron:nom_institutBVA                  0.75768    1.75014
## candidatEric Zemmour:nom_institutBVA                    -0.46013    1.75014
## candidatFabien Roussel:nom_institutBVA                   1.43661    1.75014
## candidatJean-Luc Mélenchon:nom_institutBVA              -4.47432    1.75014
## candidatMarine Le Pen:nom_institutBVA                   -1.61439    1.75014
## candidatValérie Pécresse:nom_institutBVA                 5.43117    1.75014
## candidatYannick Jadot:nom_institutBVA                    0.47709    1.75014
## candidatEmmanuel Macron:nom_institutCluster17            0.21696    1.72049
## candidatEric Zemmour:nom_institutCluster17               0.93708    1.72049
## candidatFabien Roussel:nom_institutCluster17             1.75386    1.72049
## candidatJean-Luc Mélenchon:nom_institutCluster17        -1.72026    1.72049
## candidatMarine Le Pen:nom_institutCluster17             -2.63348    1.72049
## candidatValérie Pécresse:nom_institutCluster17           5.22876    1.72049
## candidatYannick Jadot:nom_institutCluster17              0.59139    1.72049
## candidatEmmanuel Macron:nom_institutElabe                1.38103    1.71424
```

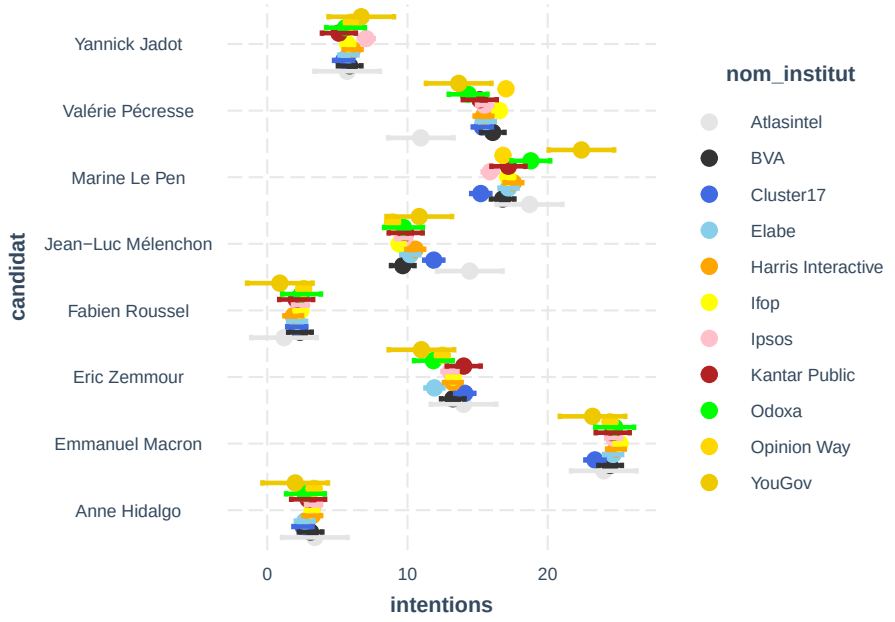
```
## candidatEric Zemmour:nom_institutElabe                    -1.35629    1.71424
## candidatFabien Roussel:nom_institutElabe                   1.64477    1.71424
## candidatJean-Luc Mélenchon:nom_institutElabe              -3.52444    1.71424
## candidatMarine Le Pen:nom_institutElabe                   -0.78677    1.71424
## candidatValérie Pécresse:nom_institutElabe                 5.34773    1.71424
## candidatYannick Jadot:nom_institutElabe                    0.80139    1.71424
## candidatEmmanuel Macron:nom_institutHarris Interactive     1.05598    1.71531
## candidatEric Zemmour:nom_institutHarris Interactive       -0.57494    1.71531
## candidatFabien Roussel:nom_institutHarris Interactive      0.83821    1.71531
## candidatJean-Luc Mélenchon:nom_institutHarris Interactive -3.70231    1.71531
## candidatMarine Le Pen:nom_institutHarris Interactive      -0.96863    1.71531
## candidatValérie Pécresse:nom_institutHarris Interactive    4.65034    1.71531
## candidatYannick Jadot:nom_institutHarris Interactive       0.56962    1.71531
## candidatEmmanuel Macron:nom_institutIfop                   1.34434    1.67547
## candidatEric Zemmour:nom_institutIfop                     -0.55201    1.67547
## candidatFabien Roussel:nom_institutIfop                    1.38693    1.67547
## candidatJean-Luc Mélenchon:nom_institutIfop               -4.84893    1.67547
## candidatMarine Le Pen:nom_institutIfop                    -1.37146    1.67547
## candidatValérie Pécresse:nom_institutIfop                  5.78598    1.67547
## candidatYannick Jadot:nom_institutIfop                     0.22809    1.67547
## candidatEmmanuel Macron:nom_institutIpsos                  0.79936    1.68345
## candidatEric Zemmour:nom_institutIpsos                    -0.87532    1.68345
## candidatFabien Roussel:nom_institutIpsos                   1.26446    1.68345
## candidatJean-Luc Mélenchon:nom_institutIpsos              -4.62283    1.68345
## candidatMarine Le Pen:nom_institutIpsos                   -2.71909    1.68345
## candidatValérie Pécresse:nom_institutIpsos                 4.63358    1.68345
## candidatYannick Jadot:nom_institutIpsos                    1.44225    1.68345
## candidatEmmanuel Macron:nom_institutKantar Public          1.11499    1.85577
## candidatEric Zemmour:nom_institutKantar Public             0.49465    1.85577
## candidatFabien Roussel:nom_institutKantar Public           1.34180    1.85577
## candidatJean-Luc Mélenchon:nom_institutKantar Public      -4.09037    1.85577
## candidatMarine Le Pen:nom_institutKantar Public           -1.02748    1.85577
## candidatValérie Pécresse:nom_institutKantar Public         4.67986    1.85577
## candidatYannick Jadot:nom_institutKantar Public           -0.11224    1.85577
## candidatEmmanuel Macron:nom_institutOdoxa                  1.45332    1.91734
## candidatEric Zemmour:nom_institutOdoxa                    -1.47379    1.91734
## candidatFabien Roussel:nom_institutOdoxa                   1.92240    1.91734
## candidatJean-Luc Mélenchon:nom_institutOdoxa              -4.05383    1.91734
## candidatMarine Le Pen:nom_institutOdoxa                    0.76336    1.91734
## candidatValérie Pécresse:nom_institutOdoxa                 4.03981    1.91734
## candidatYannick Jadot:nom_institutOdoxa                    0.55035    1.91734
## candidatEmmanuel Macron:nom_institutOpinion Way            0.50026    1.67056
## candidatEric Zemmour:nom_institutOpinion Way              -1.46076    1.67056
## candidatFabien Roussel:nom_institutOpinion Way             1.44706    1.67056
## candidatJean-Luc Mélenchon:nom_institutOpinion Way        -5.45426    1.67056
## candidatMarine Le Pen:nom_institutOpinion Way             -1.84075    1.67056
```

```
## candidatValérie Pécresse:nom_institutOpinion Way          6.12136    1.67056
## candidatYannick Jadot:nom_institutOpinion Way             0.33856    1.67056
## candidatEmmanuel Macron:nom_institutYouGov                0.60000    2.34608
## candidatEric Zemmour:nom_institutYouGov                  -1.60000    2.34608
## candidatFabien Roussel:nom_institutYouGov                 1.10000    2.34608
## candidatJean-Luc Mélenchon:nom_institutYouGov            -2.20000    2.34608
## candidatMarine Le Pen:nom_institutYouGov                  5.10000    2.34608
## candidatValérie Pécresse:nom_institutYouGov               4.10000    2.34608
## candidatYannick Jadot:nom_institutYouGov                  2.40000    2.34608
##                                                          t value Pr(>|t|)
## (Intercept)                                                2.853 0.004377 **
## candidatEmmanuel Macron                                   12.244  < 2e-16 ***
## candidatEric Zemmour                                       6.299 3.66e-10 ***
## candidatFabien Roussel                                    -1.305 0.192163
## candidatJean-Luc Mélenchon                                 6.565 6.59e-11 ***
## candidatMarine Le Pen                                      9.099  < 2e-16 ***
## candidatValérie Pécresse                                   4.492 7.44e-06 ***
## candidatYannick Jadot                                      1.372 0.170242
## tps2                                                      -3.158 0.001611 **
## tps3                                                      -5.009 5.96e-07 ***
## nom_institutBVA                                           -0.244 0.807081
## nom_institutCluster17                                     -0.695 0.487241
## nom_institutElabe                                         -0.590 0.555298
## nom_institutHarris Interactive                            -0.151 0.879779
## nom_institutIfop                                          -0.154 0.877598
## nom_institutIpsos                                         -0.075 0.940383
## nom_institutKantar Public                                 -0.364 0.715551
## nom_institutOdoxa                                         -0.495 0.620703
## nom_institutOpinion Way                                   -0.042 0.966747
## nom_institutYouGov                                        -0.844 0.398816
## candidatEmmanuel Macron:tps2                               2.424 0.015445 *
## candidatEric Zemmour:tps2                                  6.961 4.55e-12 ***
## candidatFabien Roussel:tps2                                6.977 4.07e-12 ***
## candidatJean-Luc Mélenchon:tps2                            5.625 2.11e-08 ***
## candidatMarine Le Pen:tps2                                 1.968 0.049184 *
## candidatValérie Pécresse:tps2                             -3.206 0.001367 **
## candidatYannick Jadot:tps2                                -0.522 0.601458
## candidatEmmanuel Macron:tps3                              17.098  < 2e-16 ***
## candidatEric Zemmour:tps3                                 -3.557 0.000383 ***
## candidatFabien Roussel:tps3                                7.464 1.24e-13 ***
## candidatJean-Luc Mélenchon:tps3                           18.363  < 2e-16 ***
## candidatMarine Le Pen:tps3                                12.791  < 2e-16 ***
## candidatValérie Pécresse:tps3                            -16.597  < 2e-16 ***
## candidatYannick Jadot:tps3                                 1.039 0.298749
## candidatEmmanuel Macron:nom_institutBVA                    0.433 0.665115
## candidatEric Zemmour:nom_institutBVA                      -0.263 0.792645
```

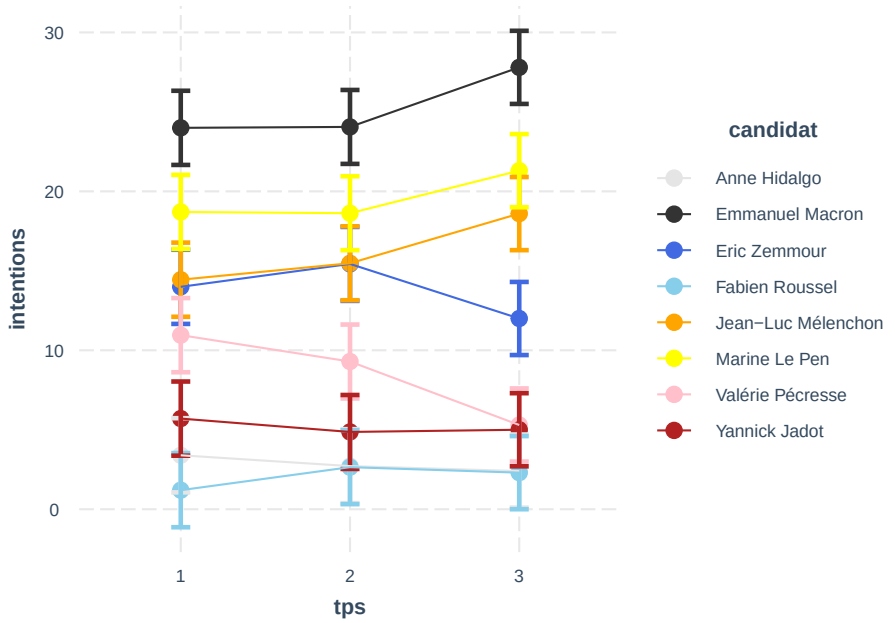
```
## candidatFabien Roussel:nom_institutBVA                      0.821 0.411826
## candidatJean-Luc Mélenchon:nom_institutBVA                 -2.557 0.010644 *
## candidatMarine Le Pen:nom_institutBVA                      -0.922 0.356411
## candidatValérie Pécresse:nom_institutBVA                    3.103 0.001940 **
## candidatYannick Jadot:nom_institutBVA                       0.273 0.785186
## candidatEmmanuel Macron:nom_institutCluster17               0.126 0.899664
## candidatEric Zemmour:nom_institutCluster17                  0.545 0.586049
## candidatFabien Roussel:nom_institutCluster17                1.019 0.308139
## candidatJean-Luc Mélenchon:nom_institutCluster17           -1.000 0.317496
## candidatMarine Le Pen:nom_institutCluster17                -1.531 0.126012
## candidatValérie Pécresse:nom_institutCluster17              3.039 0.002403 **
## candidatYannick Jadot:nom_institutCluster17                 0.344 0.731085
## candidatEmmanuel Macron:nom_institutElabe                   0.806 0.420553
## candidatEric Zemmour:nom_institutElabe                     -0.791 0.428924
## candidatFabien Roussel:nom_institutElabe                    0.959 0.337434
## candidatJean-Luc Mélenchon:nom_institutElabe               -2.056 0.039913 *
## candidatMarine Le Pen:nom_institutElabe                    -0.459 0.646309
## candidatValérie Pécresse:nom_institutElabe                  3.120 0.001837 **
## candidatYannick Jadot:nom_institutElabe                     0.467 0.640200
## candidatEmmanuel Macron:nom_institutHarris Interactive      0.616 0.538214
## candidatEric Zemmour:nom_institutHarris Interactive        -0.335 0.737521
## candidatFabien Roussel:nom_institutHarris Interactive       0.489 0.625133
## candidatJean-Luc Mélenchon:nom_institutHarris Interactive -2.158 0.031015 *
## candidatMarine Le Pen:nom_institutHarris Interactive       -0.565 0.572343
## candidatValérie Pécresse:nom_institutHarris Interactive     2.711 0.006763 **
## candidatYannick Jadot:nom_institutHarris Interactive        0.332 0.739863
## candidatEmmanuel Macron:nom_institutIfop                    0.802 0.422437
## candidatEric Zemmour:nom_institutIfop                      -0.329 0.741837
## candidatFabien Roussel:nom_institutIfop                     0.828 0.407892
## candidatJean-Luc Mélenchon:nom_institutIfop                -2.894 0.003844 **
## candidatMarine Le Pen:nom_institutIfop                     -0.819 0.413138
## candidatValérie Pécresse:nom_institutIfop                   3.453 0.000565 ***
## candidatYannick Jadot:nom_institutIfop                      0.136 0.891729
## candidatEmmanuel Macron:nom_institutIpsos                   0.475 0.634955
## candidatEric Zemmour:nom_institutIpsos                     -0.520 0.603149
## candidatFabien Roussel:nom_institutIpsos                    0.751 0.452670
## candidatJean-Luc Mélenchon:nom_institutIpsos               -2.746 0.006085 **
## candidatMarine Le Pen:nom_institutIpsos                    -1.615 0.106425
## candidatValérie Pécresse:nom_institutIpsos                  2.752 0.005968 **
## candidatYannick Jadot:nom_institutIpsos                     0.857 0.391697
## candidatEmmanuel Macron:nom_institutKantar Public           0.601 0.548025
## candidatEric Zemmour:nom_institutKantar Public              0.267 0.789843
## candidatFabien Roussel:nom_institutKantar Public            0.723 0.469738
## candidatJean-Luc Mélenchon:nom_institutKantar Public       -2.204 0.027628 *
## candidatMarine Le Pen:nom_institutKantar Public            -0.554 0.579867
## candidatValérie Pécresse:nom_institutKantar Public          2.522 0.011753 *
```

```
## candidatYannick Jadot:nom_institutKantar Public            -0.060 0.951779
## candidatEmmanuel Macron:nom_institutOdoxa                   0.758 0.448546
## candidatEric Zemmour:nom_institutOdoxa                     -0.769 0.442182
## candidatFabien Roussel:nom_institutOdoxa                    1.003 0.316155
## candidatJean-Luc Mélenchon:nom_institutOdoxa               -2.114 0.034612 *
## candidatMarine Le Pen:nom_institutOdoxa                     0.398 0.690574
## candidatValérie Pécresse:nom_institutOdoxa                  2.107 0.035242 *
## candidatYannick Jadot:nom_institutOdoxa                     0.287 0.774112
## candidatEmmanuel Macron:nom_institutOpinion Way             0.299 0.764623
## candidatEric Zemmour:nom_institutOpinion Way               -0.874 0.381997
## candidatFabien Roussel:nom_institutOpinion Way              0.866 0.386475
## candidatJean-Luc Mélenchon:nom_institutOpinion Way         -3.265 0.001113 **
## candidatMarine Le Pen:nom_institutOpinion Way              -1.102 0.270647
## candidatValérie Pécresse:nom_institutOpinion Way            3.664 0.000254 ***
## candidatYannick Jadot:nom_institutOpinion Way               0.203 0.839420
## candidatEmmanuel Macron:nom_institutYouGov                  0.256 0.798174
## candidatEric Zemmour:nom_institutYouGov                    -0.682 0.495325
## candidatFabien Roussel:nom_institutYouGov                   0.469 0.639216
## candidatJean-Luc Mélenchon:nom_institutYouGov              -0.938 0.348494
## candidatMarine Le Pen:nom_institutYouGov                    2.174 0.029834 *
## candidatValérie Pécresse:nom_institutYouGov                 1.748 0.080687 .
## candidatYannick Jadot:nom_institutYouGov                    1.023 0.306439
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.173 on 2016 degrees of freedom
## Multiple R-squared:  0.9788, Adjusted R-squared:  0.9777
## F-statistic: 902.8 on 103 and 2016 DF,  p-value: < 2.2e-16
```

```
library(jtools)

library(interactions)
cat_plot(fit2, pred=candidat,modx= nom_institut, color.class="Spectral")+
  scale_color_manual(values = SensiP2)+coord_flip()
```




```
cat_plot(fit2, pred= tps,modx=candidat, color.class="Spectral", dodge.width=0)+
  scale_color_manual(values = SensiP2)+geom_line(aes(color=candidat))
```



Chapter 7

Analyses factorielles

7.1 Origine et histoire

Par analyse factorielle, on entend finalement un ensemble de méthodes dont l'objectifs est d'extraire d'un ensemble multivariée de données, un petit nombre de dimensions, les facteurs, qui rendent compte l'essentiel des variations. Elles partagent aussi une même structure mathématique qui permet de décomposer et de réduire une matrice de données en un ensemble de matrice de dimensions réduite.

On peut en distinguer deux écoles, l'une alimentée par des questions de psychométrie a nourrit plusieurs decennies de recherche en traitant les tests psychométriques. L'autre française s'intéressent aux variables qualitatives, et a une perspective plus descriptive.

7.1.1 Une petite histoire de la psychométrie

L'analyse factorielle trouve son origine, en psychologie, dans l'intuition que dans des épreuves multiples un facteur principal contrôle les variation des items (les performance à différents tests). Mais c'est avec Thurstone que l'idée prend toute son ampleur en permettant que plusieurs facteurs traduisent la structure de la matrice de corrélations entre les tests. Spearman, hotelling,.

Dans le monde de la gestion et en particulier de la GRH et du marketing, largement inspirés par la psychologie et la psychologie sociale, ces méthodes se sont propagées et ont formalisé un processus d'étude largement fondé sur ces techniques. Il est bien connu par de processus de Churchill qui a synthésisé une manière de construire et de développé des instruments de mesure par questionnaire. C'est l' article de historique de Churchill (ref).