R Notebook Projet 6

ETUDE DE L'EVOLUTION DU CHIFFRE D'AFFAIRES ET DES RELATIONS PRODUITS ET CLIENTS / LIBRAIRIE LAPAGE

Logo Lapage

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.0     ✓ readr     2.1.4
## ✓ forcats   1.0.0     ✓ stringr   1.5.0
## ✓ ggplot2   3.4.1     ✓ tibble    3.1.8
## ✓ lubridate 1.9.2     ✓ tidyr     1.3.0
## ✓ purrr     1.0.1
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the ]8;;http://conflicted.r-lib.org/conflicted package]8;; to force all conflicts to
become errors
```

PREPARATION ET EXPLORATION DES DONNEES

```
customers  <-  read_csv("customers.csv", col_names = TRUE)
```

```
## Rows: 8623 Columns: 3
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (2): client_id, sex
## dbl (1): birth
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(customers)
```

client_id <chr>	sex <chr>	birth <dbl>
c_4410	f	1967
c_7839	f	1975
c_1699	f	1984
c_5961	f	1962
c_5320	m	1943
c_415	m	1993

6 rows

```
summary(customers)
```

```
##   client_id             sex                birth
##  Length:8623        Length:8623        Min.   :1929
##  Class :character   Class :character   1st Qu.:1966
##  Mode  :character   Mode  :character   Median :1979
##                                        Mean   :1978
##                                        3rd Qu.:1992
##                                        Max.   :2004
```

```
sum(duplicated(customers)) # pas de doublon
```

```
## [1] 0
```

```
products <- read_csv("products.csv", col_names = TRUE)
```

```
## Rows: 3287 Columns: 3
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (1): id_prod
## dbl (2): price, categ
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(products)
```

id_prod <chr>	price <dbl>	categ <dbl>
0_1421	19.99	0
0_1368	5.13	0
0_731	17.99	0
1_587	4.99	1
0_1507	3.99	0
0_1163	9.99	0

6 rows

```
summary(products)
```

```
##    id_prod             price            categ
##  Length:3287        Min.   : -1.00   Min.   :0.0000
##  Class :character   1st Qu.:  6.99   1st Qu.:0.0000
##  Mode  :character   Median : 13.06   Median :0.0000
##                     Mean   : 21.86   Mean   :0.3702
##                     3rd Qu.: 22.99   3rd Qu.:1.0000
##                     Max.   :300.00   Max.   :2.0000
```

Un ou plusieurs porduits ont un prix négatif. Examen.

```
products %>%
  filter(price == -1)
```

id_prod <chr>	price <dbl>	categ <dbl>
T_0	-1	0

1 row

Un seul produit concerné, T_0. Avoir ou autre ? A noter.

```
sum(duplicated(products)) # pas de doublon
```

```
## [1] 0
```

```
transactions <- read_csv("transactions.csv", col_names = TRUE)
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

```
## Rows: 679532 Columns: 4
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (3): id_prod, session_id, client_id
## dttm (1): date
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(transactions)
```

id_prod <chr>	date <dttm>	session_id <chr>	client_id <chr>
0_1518	2022-05-20 13:21:29.043970	s_211425	c_103
1_251	2022-02-02 07:55:19.149409	s_158752	c_8534
0_1277	2022-06-18 15:44:33.155328	s_225667	c_6714
2_209	2021-06-24 04:19:29.835891	s_52962	c_6941
0_1509	2023-01-11 08:22:08.194478	s_325227	c_4232
0_1418	2022-10-20 15:59:16.084029	s_285425	c_1478

6 rows

```
summary(transactions)
```

```
##    id_prod               date                          session_id
##  Length:679532      Min.   :2021-03-01 00:01:07.83   Length:679532
##  Class :character   1st Qu.:2021-09-08 09:14:25.05   Class :character
##  Mode  :character   Median :2022-03-03 07:50:20.81   Mode  :character
##                     Mean   :2022-03-03 15:13:19.30
##                     3rd Qu.:2022-08-30 23:57:08.55
##                     Max.   :2023-02-28 23:58:30.78
##                     NA's   :200
##   client_id
##  Length:679532
##  Class :character
##  Mode  :character
##
##
##
##
```

Il manque 200 dates dans les données. Pourquoi ?

```
problems(transactions) # retourne les valeurs posant problèmes de traitement / package {vroo
m}
```

row <int>	col <int>	expected <chr>	actual <chr>
3021	2	date in ISO8601	test_2021-03-01 02:30:02.237419
5140	2	date in ISO8601	test_2021-03-01 02:30:02.237425
9670	2	date in ISO8601	test_2021-03-01 02:30:02.237437
10730	2	date in ISO8601	test_2021-03-01 02:30:02.237436
15294	2	date in ISO8601	test_2021-03-01 02:30:02.237430
19314	2	date in ISO8601	test_2021-03-01 02:30:02.237449
23680	2	date in ISO8601	test_2021-03-01 02:30:02.237430
23698	2	date in ISO8601	test_2021-03-01 02:30:02.237444
27780	2	date in ISO8601	test_2021-03-01 02:30:02.237437
35435	2	date in ISO8601	test_2021-03-01 02:30:02.237418

1-10 of 200 rows | 1-4 of 5 columns Previous **1** 2 3 4 5 6 … 20 Next

```
filter(transactions, is.na(transactions$date))
```

id_prod <chr>	date <dttm>	session_id <chr>	client_id <chr>
T_0	<NA>	s_0	ct_0

id_prod <chr>	date <dttm>	session_id <chr>	client_id <chr>
T_0	<NA>	s_0	ct_0
T_0	<NA>	s_0	ct_1
T_0	<NA>	s_0	ct_0
T_0	<NA>	s_0	ct_0
T_0	<NA>	s_0	ct_0
T_0	<NA>	s_0	ct_1
T_0	<NA>	s_0	ct_0
T_0	<NA>	s_0	ct_1
T_0	<NA>	s_0	ct_1

1-10 of 200 rows Previous **1** 2 3 4 5 6 … 20 Next

```
filter(transactions, transactions$id_prod == "T_0")
```

id_prod <chr>	date <dttm>	session_id <chr>	client_id <chr>
T_0	<NA>	s_0	ct_0
T_0	<NA>	s_0	ct_0
T_0	<NA>	s_0	ct_1
T_0	<NA>	s_0	ct_0
T_0	<NA>	s_0	ct_0
T_0	<NA>	s_0	ct_0
T_0	<NA>	s_0	ct_1
T_0	<NA>	s_0	ct_0
T_0	<NA>	s_0	ct_1
T_0	<NA>	s_0	ct_1

1-10 of 200 rows Previous **1** 2 3 4 5 6 … 20 Next

Visiblement, il s'agit d'un test ayant eu lieu à la même date et portant la mention "test_2021-03-01". Les lignes sont à supprimer.

```
transactions <- transactions %>%
  filter(transactions$id_prod != "T_0")

summary(transactions)  # 679532 - 679332 = 200 , le compte est bon
```

```
##    id_prod              date                         session_id
##  Length:679332      Min.   :2021-03-01 00:01:07.83   Length:679332
##  Class :character   1st Qu.:2021-09-08 09:14:25.05   Class :character
##  Mode  :character   Median :2022-03-03 07:50:20.81   Mode  :character
##                     Mean   :2022-03-03 15:13:19.30
##                     3rd Qu.:2022-08-30 23:57:08.55
##                     Max.   :2023-02-28 23:58:30.78
##   client_id
##  Length:679332
##  Class :character
##  Mode  :character
##
##
##
```

```
sum(duplicated(transactions)) # pas de doublon
```

```
## [1] 0
```

En toute rigueur, il convient également de corriger le dataframe *products* qui contient le produit T_0 associé aux tests.

```
products <- products %>%
  filter(id_prod != "T_0")
```

```
library(questionr)
freq(table(products$categ))  # nombre de produits par catégorie et pourcentages associés
```

	n <dbl>	% <dbl>	val% <dbl>
0	2308	70.2	70.2
1	739	22.5	22.5
2	239	7.3	7.3

3 rows

Fusion des données transactions et products

```
trans_prod <- transactions %>%
  tidylog::left_join(products, by = "id_prod")   # {tidylog} rend le package {dplyr} plus bav
ard
```

```
## left_join: added 2 columns (price, categ)
```

```
##            > rows only in x       221
```

```
##            > rows only in y  (     21)
```

```
##            > matched rows     679,111
```

```
##            >                 =========
```

```
##            > rows total       679,332
```

```
head(trans_prod)
```

id_prod <chr>	date <dttm>	session_id <chr>	client_id <chr>	price <dbl>	categ <dbl>
0_1518	2022-05-20 13:21:29.043970	s_211425	c_103	4.18	0
1_251	2022-02-02 07:55:19.149409	s_158752	c_8534	15.99	1
0_1277	2022-06-18 15:44:33.155328	s_225667	c_6714	7.99	0
2_209	2021-06-24 04:19:29.835891	s_52962	c_6941	69.99	2
0_1509	2023-01-11 08:22:08.194478	s_325227	c_4232	4.99	0
0_1418	2022-10-20 15:59:16.084029	s_285425	c_1478	8.57	0

6 rows

```
summary(trans_prod)
```

```
##    id_prod              date                         session_id
##  Length:679332      Min.   :2021-03-01 00:01:07.83   Length:679332
##  Class :character   1st Qu.:2021-09-08 09:14:25.05   Class :character
##  Mode  :character   Median :2022-03-03 07:50:20.81   Mode  :character
##                     Mean   :2022-03-03 15:13:19.30
##                     3rd Qu.:2022-08-30 23:57:08.55
##                     Max.   :2023-02-28 23:58:30.78
##
##   client_id            price            categ
##  Length:679332      Min.   :  0.62   Min.   :0.000
##  Class :character   1st Qu.:  8.87   1st Qu.:0.000
##  Mode  :character   Median : 13.99   Median :0.000
##                     Mean   : 17.45   Mean   :0.442
##                     3rd Qu.: 18.99   3rd Qu.:1.000
##                     Max.   :300.00   Max.   :2.000
##                     NA's   :221      NA's   :221
```

Il manque 221 prix et 221 entrées catégories.

```
trans_prod[is.na(trans_prod$categ) | is.na(trans_prod$price),] %>%
  group_by(id_prod) %>%
  count()
```

id_prod <chr>	n <int>
0_2245	221

1 row

Un seul produit est concerné : le 0_2245, à 221 reprises. Son préfixe le désignerait comme faisant partie de la catégorie 0. Attribution, à défaut, de la valeur médiane des prix des produits de catégorie 0.

```
med_price_cat_0 <-
  median(trans_prod[trans_prod$categ == 0,]$price, na.rm = TRUE)
trans_prod[is.na(trans_prod$categ), ]$categ <- 0
trans_prod[is.na(trans_prod$price), ]$price <- med_price_cat_0
```

```
table(trans_prod[trans_prod$id_prod == "0_2245", ]$price, trans_prod[trans_prod$id_prod ==
                                                                       "0_2245", ]$categ)
```

```
##
##          0
##   9.99 221
```

ANALYSE DES DONNEES

Analyse de la répartition du chiffre d'affaires

Chiffre d'affaires = somme des prix des produits vendus lors des transactions

```
CA_total <- sum(trans_prod$price)
CA_total
```

```
## [1] 11855936
```

CA par produit

```
CA_prod <- trans_prod %>%
  group_by(id_prod) %>%
  summarise(CA = sum(price, na.rm = TRUE))
head(CA_prod)
```

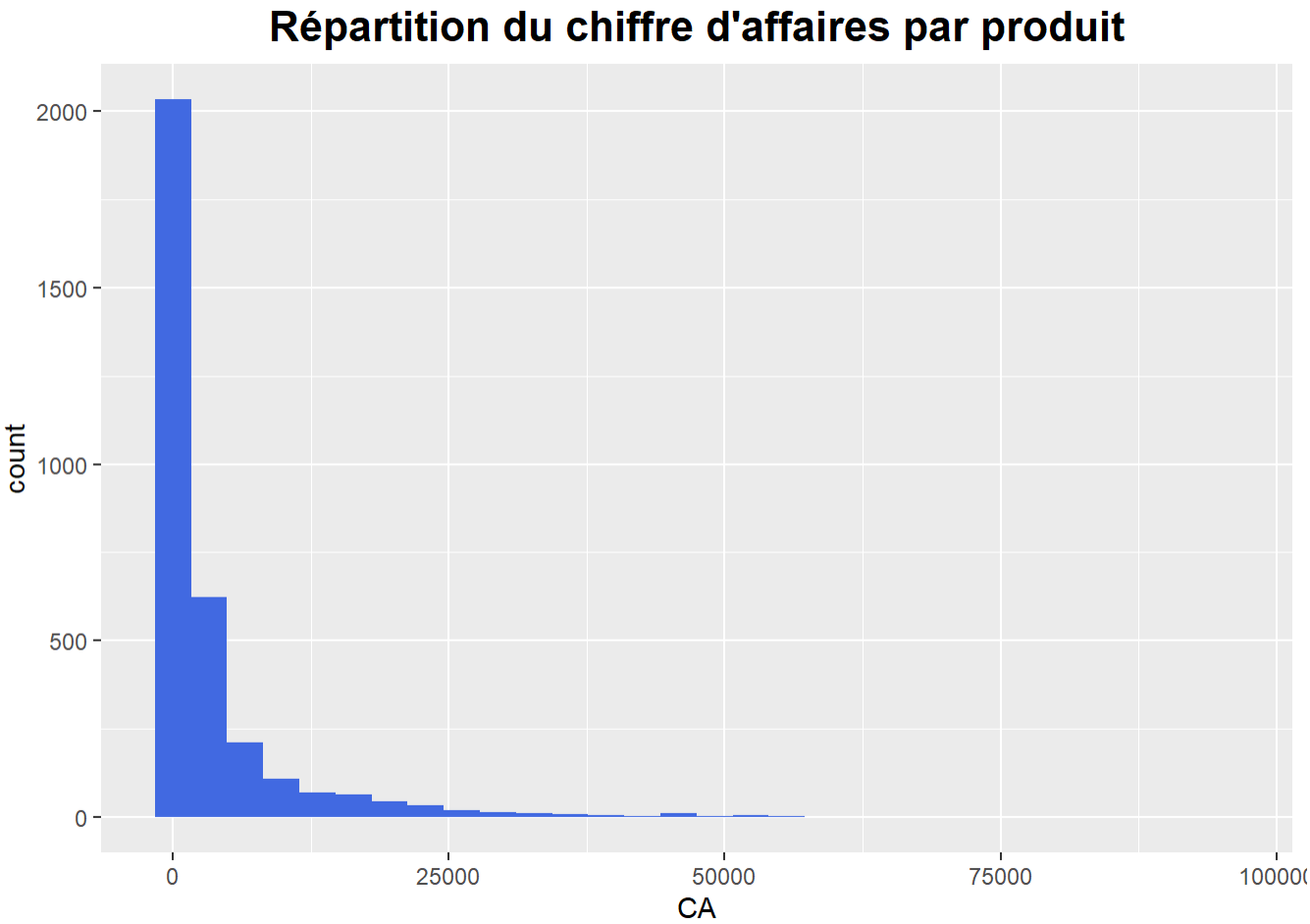
id_prod <chr>	CA <dbl>
0_0	4657.50
0_1	5352.13
0_10	394.90
0_100	61.80
0_1000	2954.88
0_1001	2020.95

6 rows

```
summary(CA_prod)
```

```
##    id_prod                CA          
##  Length:3266        Min.   :    0.99  
##  Class :character   1st Qu.:  233.83  
##  Mode  :character   Median :  797.22  
##                     Mean   : 3630.11  
##                     3rd Qu.: 3406.20  
##                     Max.   :94893.50
```

```
ggplot(CA_prod) +
  aes(x = CA) +
  geom_histogram(bins = 30L, fill = "royalblue") +
  labs(title = "Répartition du chiffre d'affaires par produit") +
  theme_grey() +
  theme(plot.title = element_text(size = 16L,
                                  face = "bold",
                                  hjust = 0.5))
```



Relevé de quelques données significatives concernant le CA des produits.

```
# Les 6 premiers produits classés par CA
head(CA_prod %>%
       arrange(desc(CA)))
```

id_prod <chr>	CA <dbl>
2_159	94893.50
2_135	69334.95
2_112	65407.76
2_102	60736.78

id_prod <chr>	CA <dbl>
2_209	56971.86
1_395	54356.25

6 rows

```
# Les 6 derniers produits pour le CA / summary() a déjà donné moyenne, médiane et valeurs extrêmes.
head(CA_prod %>%
       arrange(CA))
```

id_prod <chr>	CA <dbl>
0_1539	0.99
0_1284	1.38
0_1653	1.98
0_1601	1.99
0_541	1.99
0_807	1.99

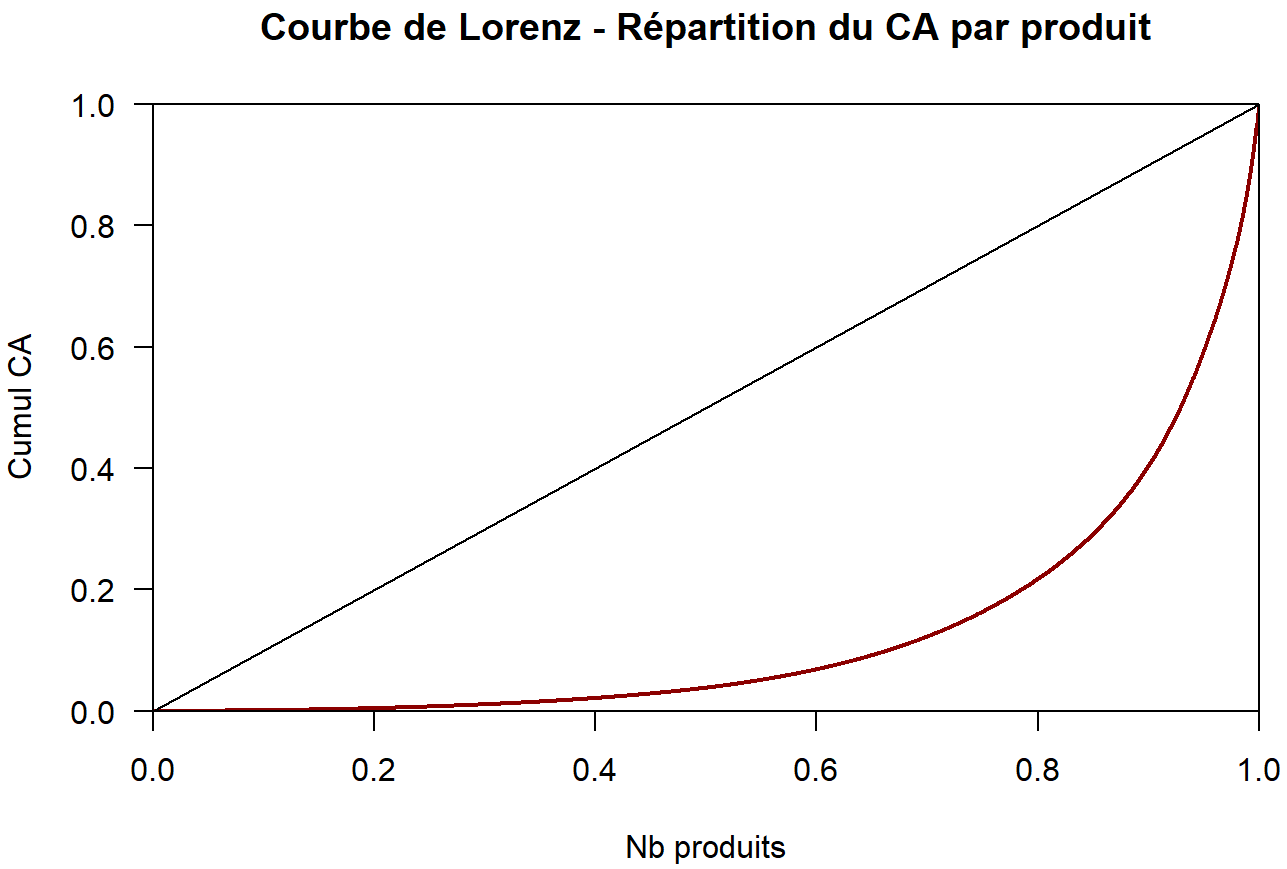
6 rows

Mesure de l'inégalité des produits en matière de CA.

```
library(ineq) # package {ineq} / Measuring Inequality, Concentration, and Poverty
```

Courbe de Lorenz de la répartition du CA par produit

```
plot(Lc(CA_prod$CA), col = "darkred", lwd = 2, main="Courbe de Lorenz - Répartition du CA par produit", xlab = "Nb produits", ylab= "Cumul CA")
```



On compare les valeurs cumulées des déciles avec la droite d'équi-répartition, c'est une estimation de l'inégalité. Ici très marquée.

Indice ou coefficient de Gini

```
ineq(CA_prod$CA, type = "Gini") # G = 2*AUC / plus G est fort, plus l'inégalité est forte. 0 = égalité parfaite, 1 inégalité parfaite jamais atteinte.
```

```
## [1] 0.7428386
```

CA par catégories

```
CA_categ  <- trans_prod %>%
  group_by(categ) %>%
  summarise(CA = sum(price),
            prop = round(sum(price) / CA_total * 100, 2))
CA_categ
```

categ <dbl>	CA <dbl>	prop <dbl>
0	4421939	37.30
1	4653723	39.25

categ	CA	prop
<dbl>	<dbl>	<dbl>
2	2780275	23.45

3 rows

Analyse de tendance et saisonnalité

Préparation des données temporelles

Regroupement des dates par mois

```
trans_prod$periode <- format(trans_prod$date, "%y-%m")
trans_prod$date_courte <- as.Date(trans_prod$date)
```

```
summary(trans_prod)
```

```
##    id_prod              date                         session_id       
##  Length:679332      Min.   :2021-03-01 00:01:07.83   Length:679332     
##  Class :character   1st Qu.:2021-09-08 09:14:25.05   Class :character  
##  Mode  :character   Median :2022-03-03 07:50:20.81   Mode  :character  
##                     Mean   :2022-03-03 15:13:19.30                     
##                     3rd Qu.:2022-08-30 23:57:08.55                     
##                     Max.   :2023-02-28 23:58:30.78                     
##   client_id             price            categ          periode         
##  Length:679332      Min.   :  0.62   Min.   :0.0000   Length:679332     
##  Class :character   1st Qu.:  8.87   1st Qu.:0.0000   Class :character  
##  Mode  :character   Median : 13.99   Median :0.0000   Mode  :character  
##                     Mean   : 17.45   Mean   :0.4418                     
##                     3rd Qu.: 18.99   3rd Qu.:1.0000                     
##                     Max.   :300.00   Max.   :2.0000                     
##   date_courte        
##  Min.   :2021-03-01  
##  1st Qu.:2021-09-08  
##  Median :2022-03-03  
##  Mean   :2022-03-03  
##  3rd Qu.:2022-08-30  
##  Max.   :2023-02-28
```

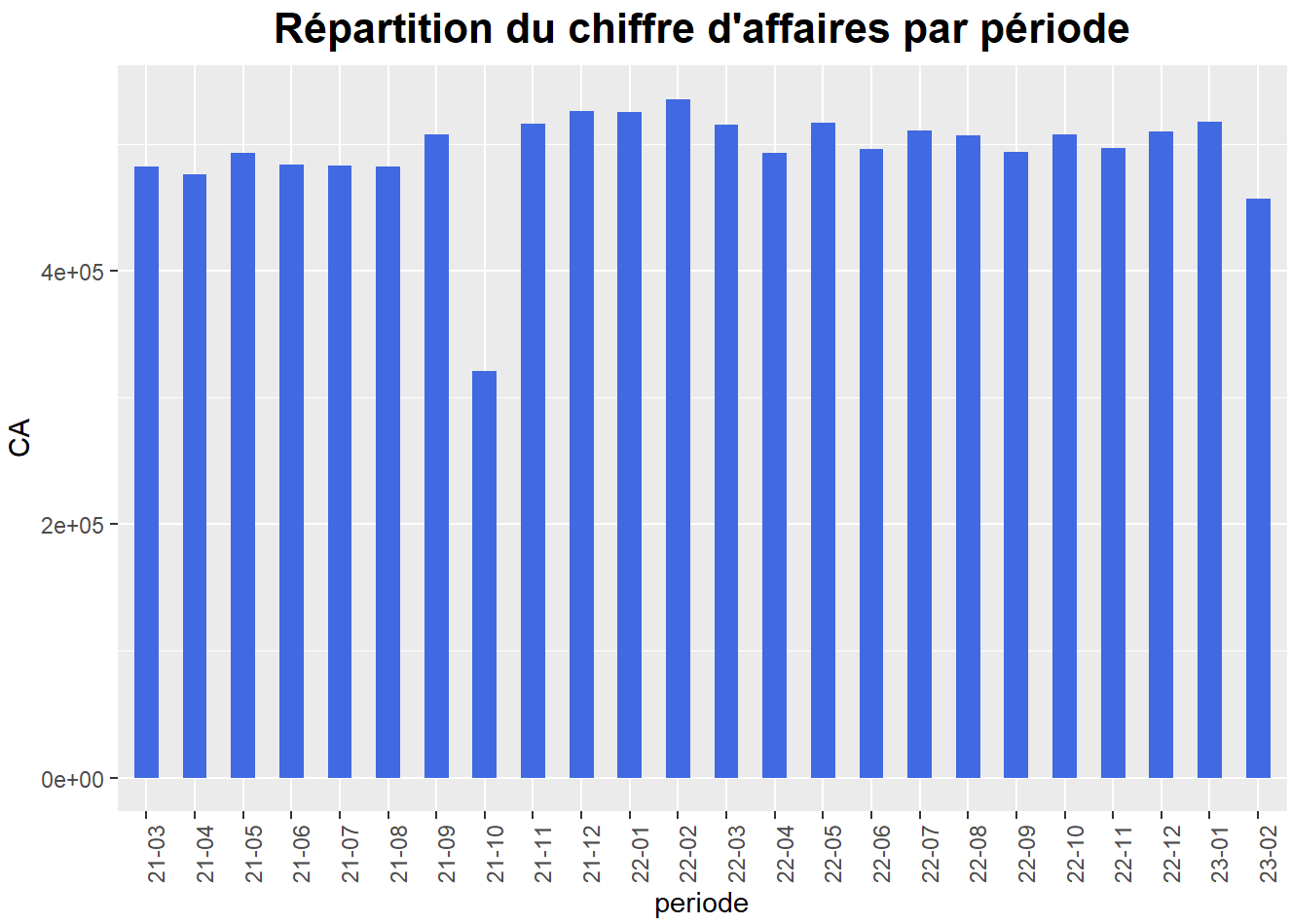
```
## Transtypage custom_prod$categ en factor
trans_prod$categ <- trans_prod$categ %>%
  as.character() %>%
  as.factor()
```

```
CA_periode  <-  trans_prod %>%
  group_by(periode) %>%
  summarise(CA = sum(price),
            prop = round(sum(price) / CA_total * 100, 2),
            count = n())
CA_periode
```

periode <chr>	CA <dbl>	prop <dbl>	count <int>
21-03	482530.5	4.07	28610
21-04	476249.2	4.02	28457
21-05	493023.4	4.16	28293
21-06	484158.5	4.08	26857
21-07	482875.4	4.07	24742
21-08	482374.7	4.07	25659
21-09	507360.6	4.28	33326
21-10	320868.7	2.71	21606
21-11	516267.6	4.35	28321
21-12	525987.2	4.44	32464

1-10 of 24 rows Previous **1** 2 3 Next

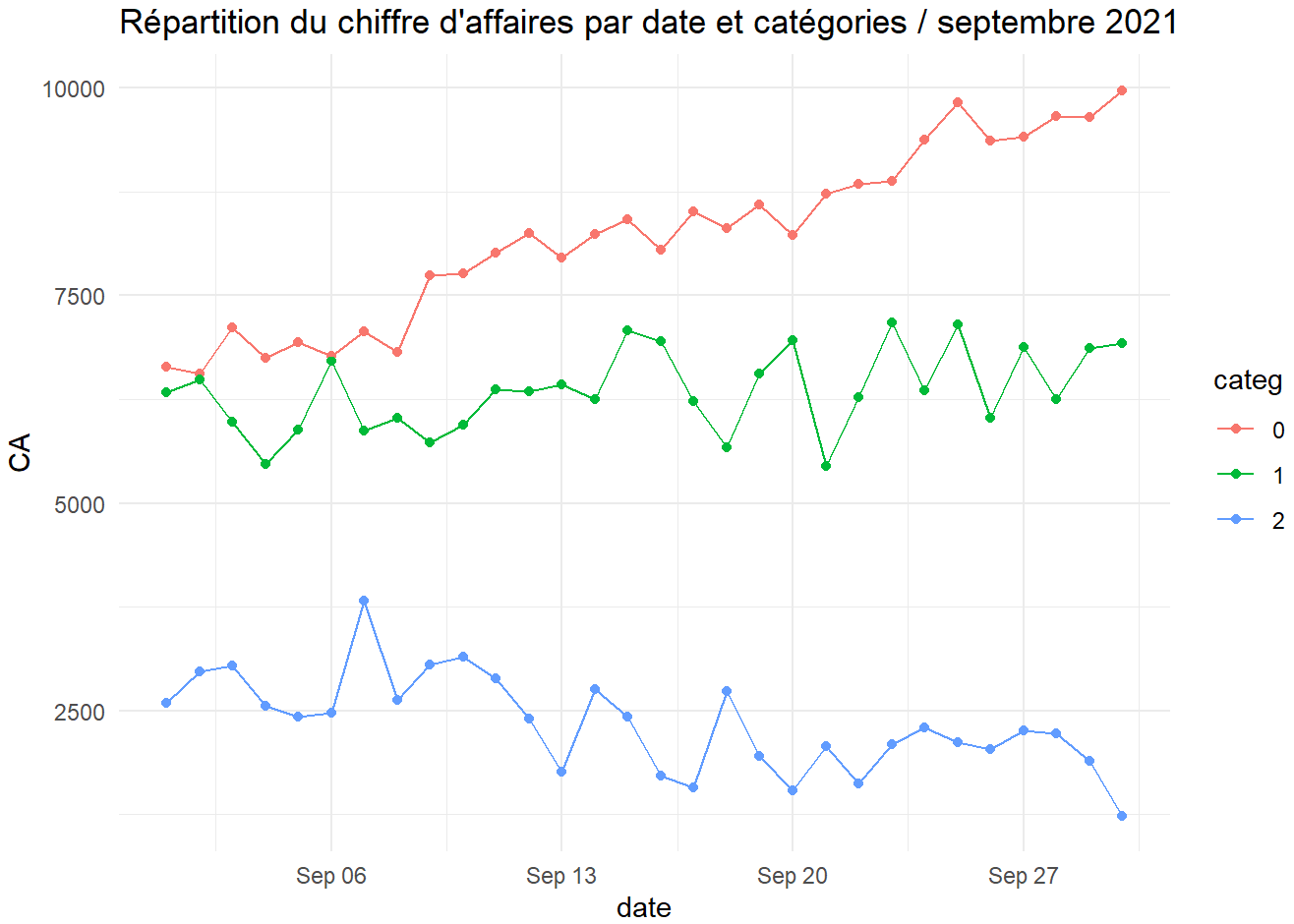
```
CA_periode %>%
  ggplot(aes(x = periode, y = CA)) +
  geom_col(width = 0.5, fill = "royalblue") +
  labs(title = "Répartition du chiffre d'affaires par période") +
  theme_grey() +
  theme(plot.title = element_text(size = 16L,
                                  face = "bold",
                                  hjust = 0.5)) +
  theme(axis.text.x = element_text(angle = 90))
```



Que s'est-il passé en octobre 2021 ?

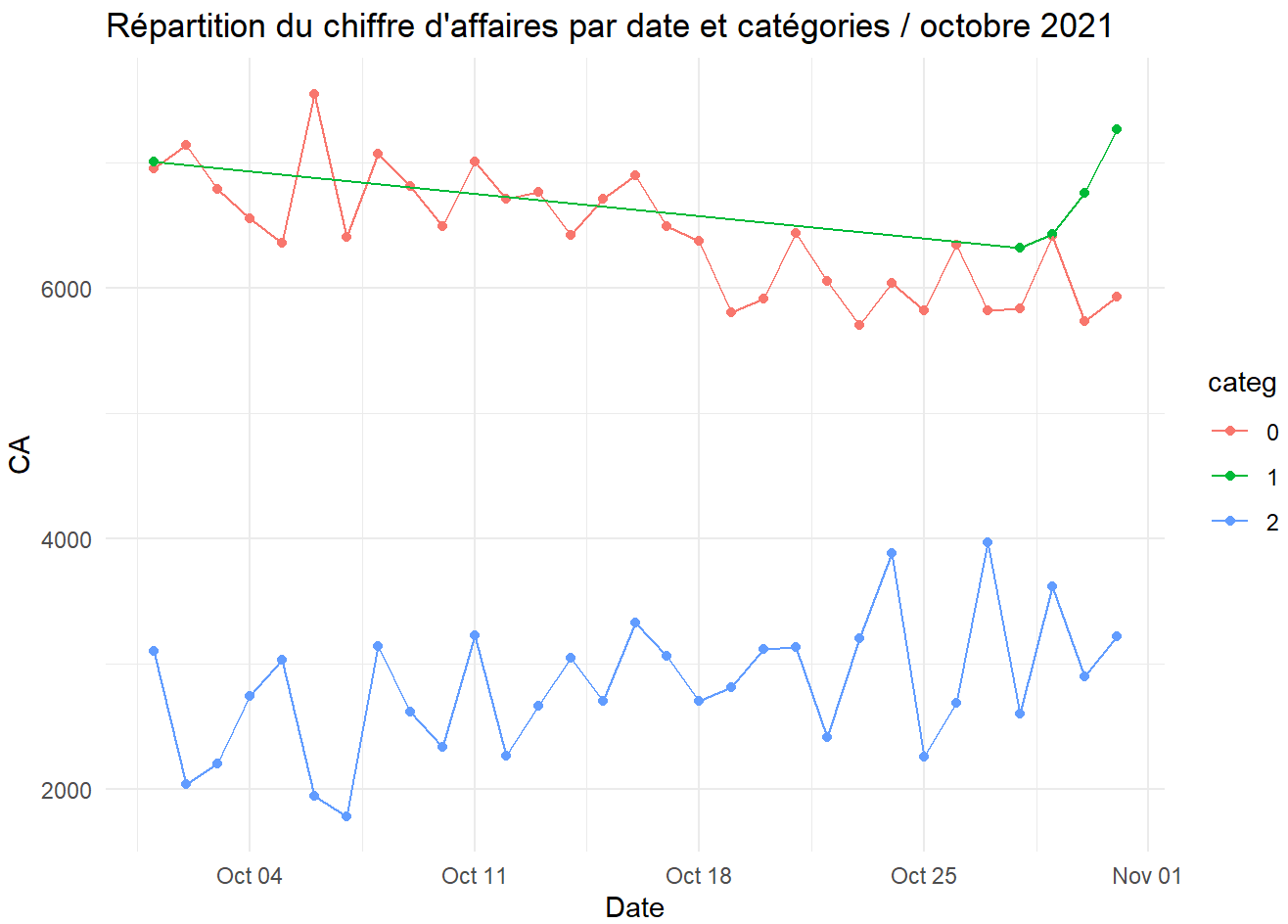
```
trans_prod[trans_prod$periode == "21-09",] %>%
  group_by(categ, date_courte) %>%
  summarise(CA = sum(price)) %>%
  ggplot(aes(x = date_courte, y = CA, colour = categ)) +
  geom_point() +
  geom_line() +
  scale_color_hue(direction = 1) +
  labs(title = "Répartition du chiffre d'affaires par date et catégories / septembre 2021") +
  labs(x = "date") +
  theme(plot.title = element_text(size = 16L,
                                  face = "bold",
                                  hjust = 0.5)) +
  theme(axis.text.x = element_text(angle = 90)) +
  theme_minimal()
```

```
## `summarise()` has grouped output by 'categ'. You can override using the
## `.groups` argument.
```




```
trans_prod[trans_prod$periode == "21-10",] %>%
  group_by(categ, date_courte) %>%
  summarise(CA = sum(price)) %>%
  ggplot(aes(x = date_courte, y = CA, colour = categ)) +
  geom_point() +
  geom_line() +
  scale_color_hue(direction = 1) +
  labs(title = "Répartition du chiffre d'affaires par date et catégories / octobre 2021") +
  labs(x = "Date") +
  theme(plot.title = element_text(size = 16L,
                                  face = "bold",
                                  hjust = 0.5)) +
  theme(axis.text.x = element_text(angle = 90)) +
  theme_minimal()
```

```
## `summarise()` has grouped output by 'categ'. You can override using the
## `.groups` argument.
```



Les ventes de livres en catégorie 1 connaissent une interruption prolongée en octobre. **C'est un évènement exceptionnel. Il convient d'en corriger l'incidence.**

```
CA_periode_corr <- CA_periode %>%
  select(-c("prop", "count"))     # simplification

CA_periode_corr[CA_periode$periode == "21-10",]$CA <-
  (CA_periode[CA_periode$periode == "21-09",]$CA + CA_periode[CA_periode$periode == "21-1
1",]$CA) /
  2
```

```
CA_periode_corr
```

periode <chr>	CA <dbl>
21-03	482530.5
21-04	476249.2
21-05	493023.4
21-06	484158.5
21-07	482875.4

periode <chr>	CA <dbl>
21-08	482374.7
21-09	507360.6
21-10	511814.1
21-11	516267.6
21-12	525987.2

1-10 of 24 rows Previous **1** 2 3 Next

Approche de la tendance et de la saisonnalité

Création de la série temporelle

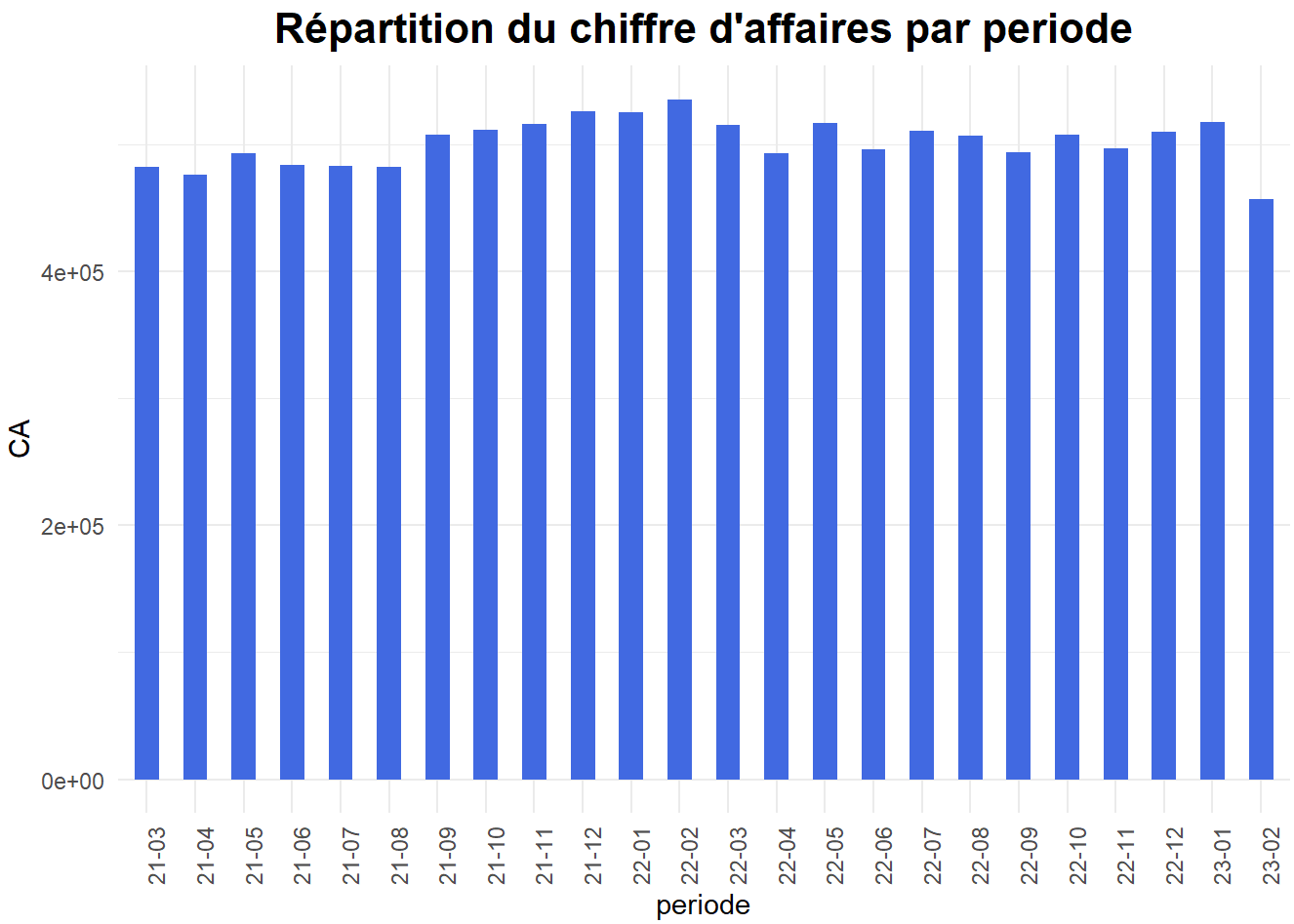
```
CA.ts <- ts(CA_periode_corr$CA, start = c(2021, 3), frequency = 12)
CA.ts
```

```
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 2021                   482530.5 476249.2 493023.4 484158.5 482875.4 482374.7
## 2022 525388.9 535681.4 515566.4 493138.8 517292.4 496086.1 510903.0 506547.2
## 2023 517610.5 456749.7
##           Sep      Oct      Nov      Dec
## 2021 507360.6 511814.1 516267.6 525987.2
## 2022 494204.4 508017.7 496774.8 510279.4
## 2023
```

```
str(CA.ts)
```

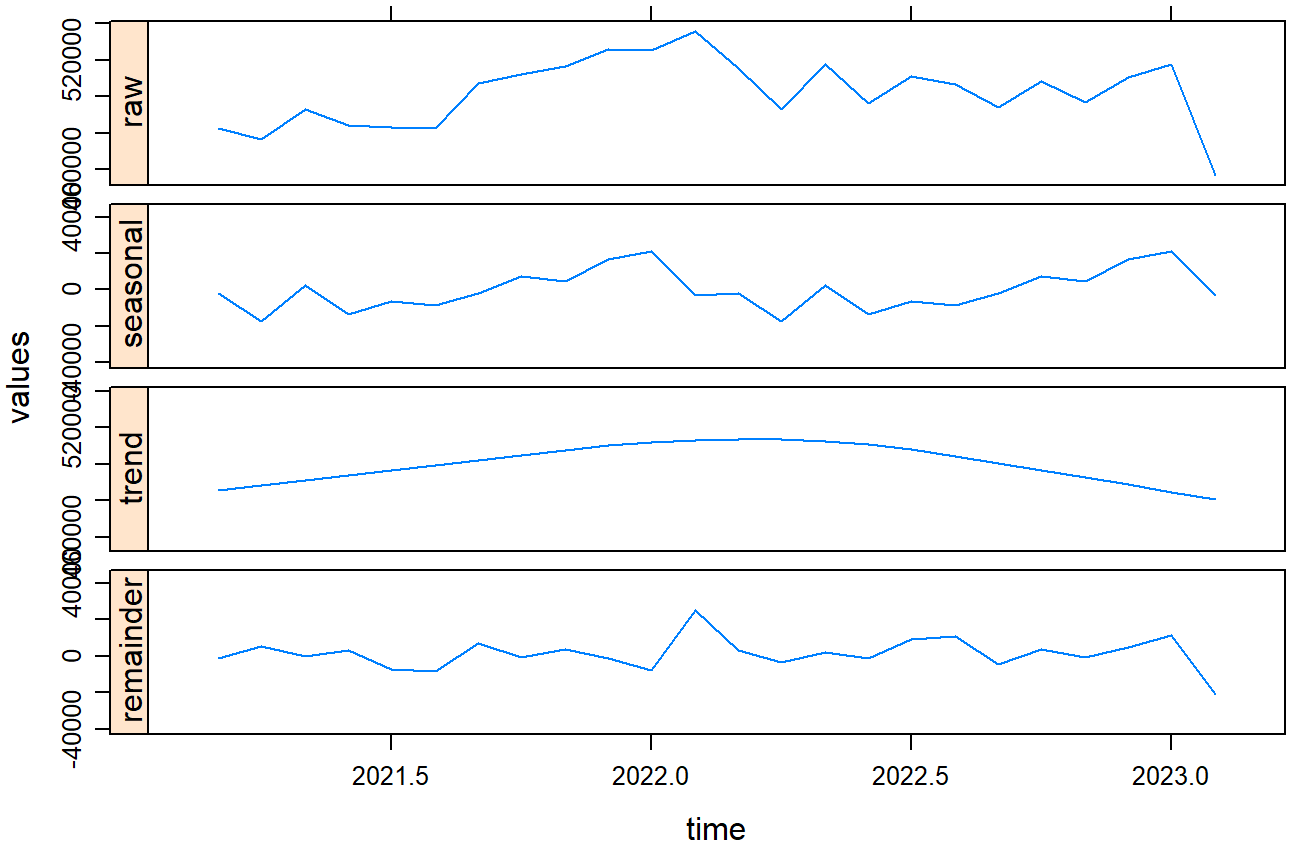
```
##  Time-Series [1:24] from 2021 to 2023: 482531 476249 493023 484158 482875 ...
```

```
CA_periode_corr %>%
  ggplot(aes(x = periode, y = CA)) +
  geom_col(width = 0.5, fill = "royalblue") +
  labs(title = "Répartition du chiffre d'affaires par periode") +
  theme_minimal() +
  theme(plot.title = element_text(size = 16L,
                                  face = "bold",
                                  hjust = 0.5)) +
  theme(axis.text.x = element_text(angle = 90))
```



Décomposition saisonnière par moyennes mobiles

```
library(stlplus)    # package {stlplus} Seasonal Decomposition of Time Series by Læss
decomp_CA_stl <- stlplus(CA.ts, s.window = "periodic")
plot(decomp_CA_stl)
```



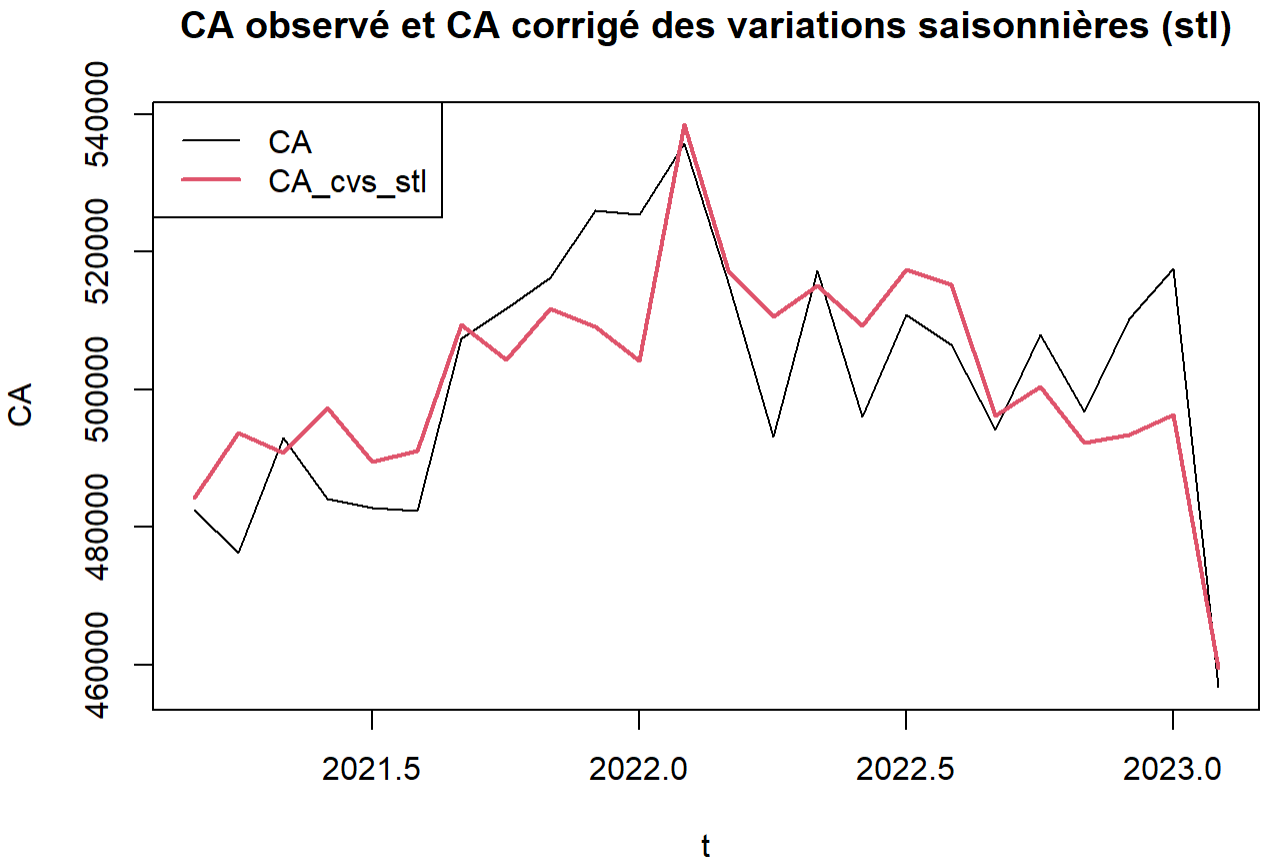
Une tendance d'abord montante en pente régulière puis descendante en pente régulière à partir du 1er trimestre 2022.

```
stl <- decomp_CA_stl$data
round(sum(stl$seasonal), 2) # valeur nulle car schéma additif
```

```
## [1] 0
```

```
CA_cvs_stl <-
  CA.ts - stl$seasonal  # schéma additif retenu par la méthode

ts.plot(
  CA.ts,
  CA_cvs_stl,
  xlab = "t",
  ylab = "CA",
  col = c(1, 2),
  lwd = c(1, 2),
  main = "CA observé et CA corrigé des variations saisonnières (stl)"
)
legend(
  "topleft",
  legend = c("CA", "CA_cvs_stl"),
  col = c(1, 2),
  lwd = c(1, 2)
)
```



Analyse de la clientèle

Détermination de l'âge / positionnement en 2023

```
customers$age  <-  2023 - customers$birth
```

```
head(customers)
```

client_id <chr>	sex <chr>	birth <dbl>	age <dbl>
c_4410	f	1967	56
c_7839	f	1975	48
c_1699	f	1984	39
c_5961	f	1962	61
c_5320	m	1943	80
c_415	m	1993	30

6 rows

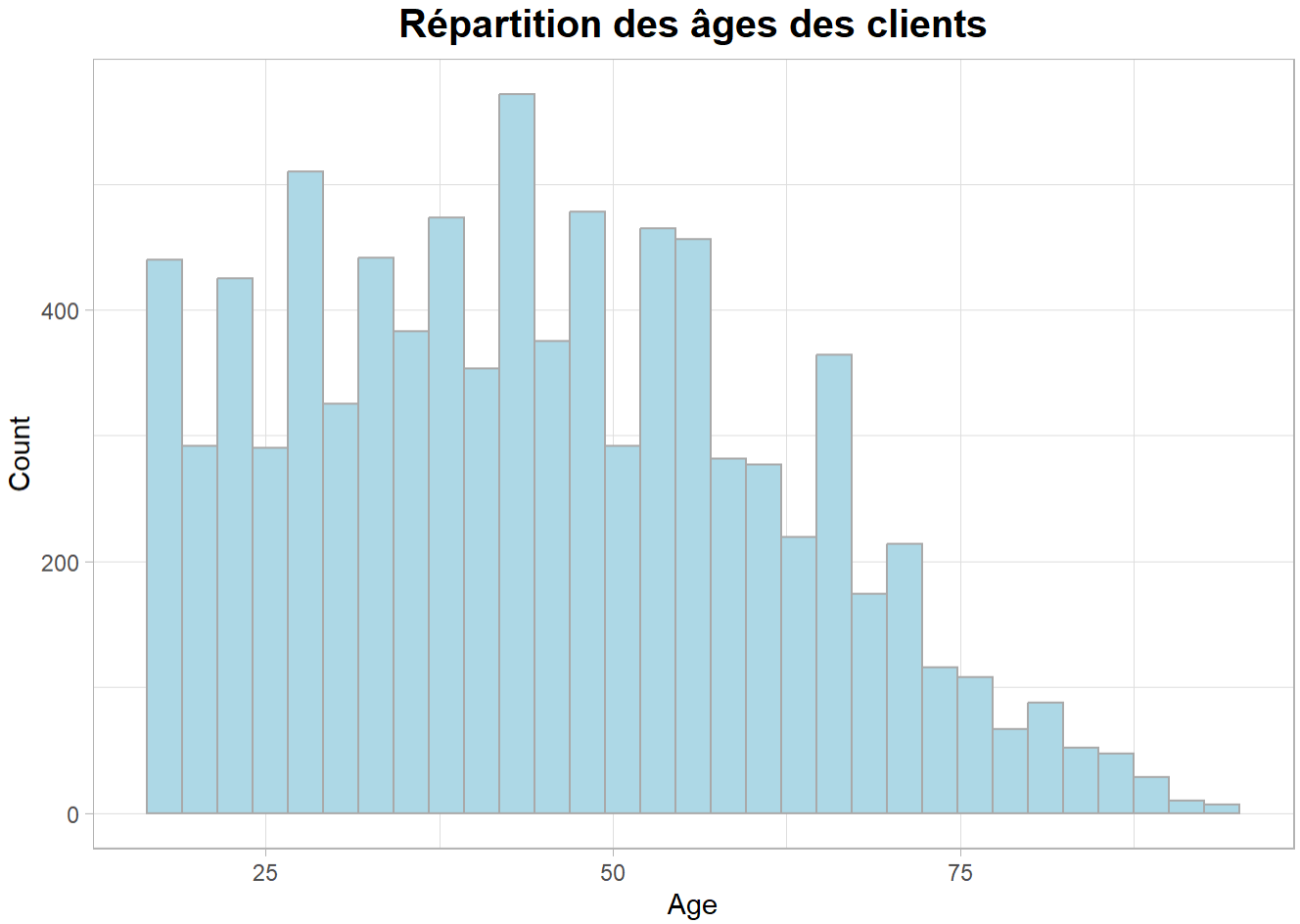
```
customers %>%
  group_by(sex) %>%
  summarise(
    age_max = max(age),
    age_min = min(age),
    age_mean = mean(age),
    age_median = median(age),
    age_sd = sd(age),
    count = n()
  )
```

sex <chr>	age_max <dbl>	age_min <dbl>	age_mean <dbl>	age_median <dbl>	age_sd <dbl>	count <int>
f	94	19	45.00512	44	17.09824	4491
m	94	19	44.40828	43	16.71966	4132

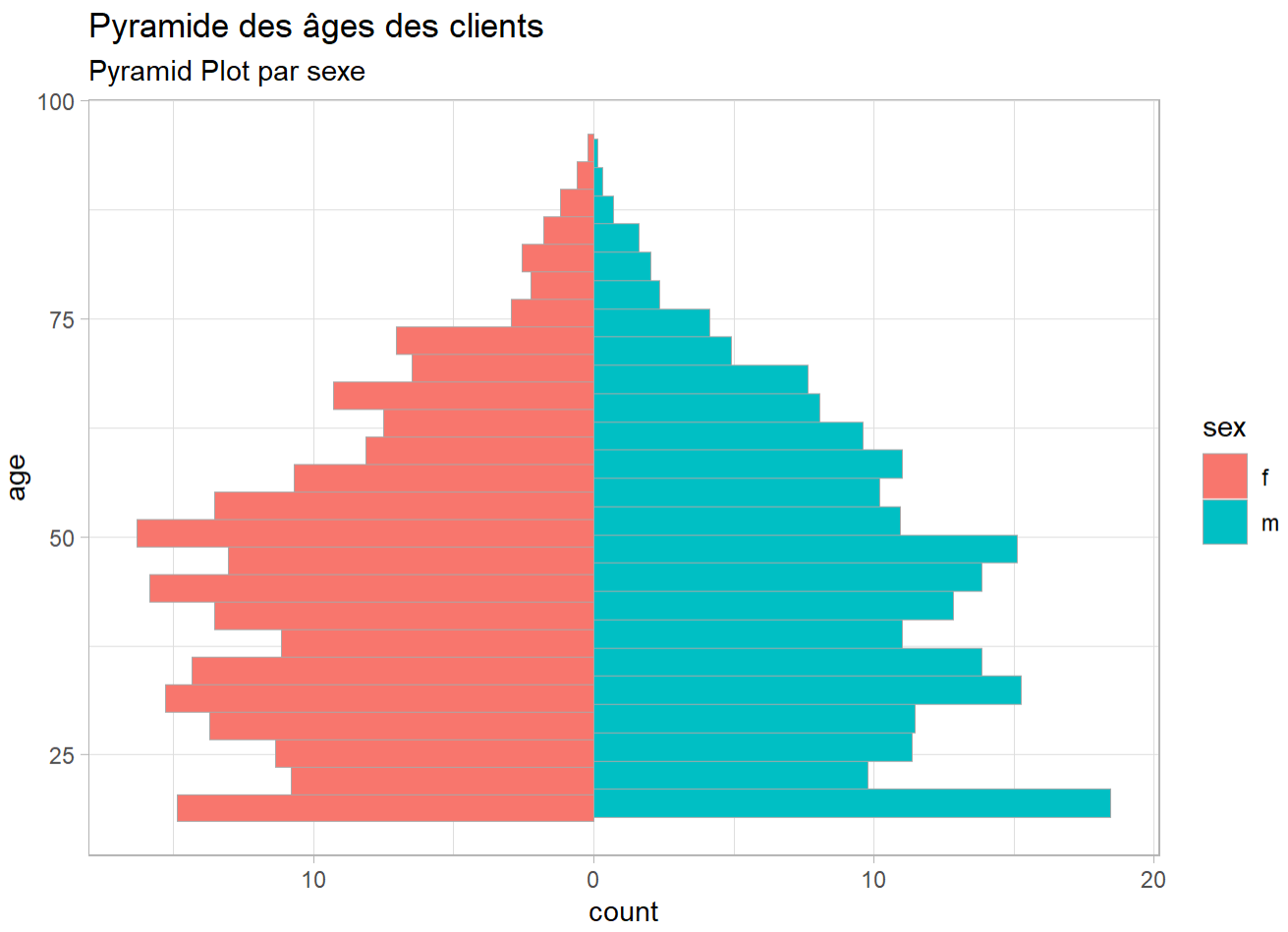
2 rows

Hommes et femmes sont très proches sur toutes les valeurs d'âges et même en nombre.

```
ggplot(customers) +
  aes(x = age) +
  geom_histogram(
    binwidth = function(x)
      2 * IQR(x) / (length(x) ^ (1 / 3)),
    fill = "lightblue",
    color = "darkgrey"
  ) +
  labs(x = "Age",
       y = "Count",
       title = "Répartition des âges des clients") +
  theme_light() +
  theme(plot.title = element_text(size = 15L,
                                  face = "bold",
                                  hjust = 0.5))
```



```
ggplot(customers, aes(fill = sex)) +
  geom_histogram(
    data = subset(customers, sex == "f"),
    linewidth = 0.1,
    binwidth = function(x)
      2 * IQR(x) / (length(x) ^ (1 / 3)),
    aes(x = age, y = after_stat(count) * (-1)),
    color = "darkgrey"
  ) +
  geom_histogram(
    data = subset(customers, sex == "m"),
    linewidth = 0.1,
    binwidth = function(x)
      2 * IQR(x) / (length(x) ^ (1 / 3)),
    aes(x = age),
    color = "darkgrey"
  ) +
  scale_y_continuous(labels = paste0(as.character(c(
    seq(20, 0, -10), seq(10, 20, 10)
  )))) +
  ylab("count") +  coord_flip() +
  labs(title = "Pyramide des âges des clients", subtitle = "Pyramid Plot par sexe") + theme_l
ight()
```



Fusion des données transactions et clients

```
custom_prod  <-  trans_prod %>%
  tidylog::left_join(customers, by = "client_id")
```

```
## left_join: added 3 columns (sex, birth, age)
```

```
##            > rows only in x         0
```

```
##            > rows only in y  (     23)
```

```
##            > matched rows     679,332
```

```
##            >                =========
```

```
##            > rows total       679,332
```

```
head(custom_prod)
```

id_prod <chr>	date <dttm>	session_id <chr>	client_id <chr>	price <dbl>	cat… <fct>	periode <chr>	date_c <c
0_1518	2022-05-20 13:21:29.043970	s_211425	c_103	4.18	0	22-05	2022-(
1_251	2022-02-02 07:55:19.149409	s_158752	c_8534	15.99	1	22-02	2022-(
0_1277	2022-06-18 15:44:33.155328	s_225667	c_6714	7.99	0	22-06	2022-(
2_209	2021-06-24 04:19:29.835891	s_52962	c_6941	69.99	2	21-06	2021-(
0_1509	2023-01-11 08:22:08.194478	s_325227	c_4232	4.99	0	23-01	2023-
0_1418	2022-10-20 15:59:16.084029	s_285425	c_1478	8.57	0	22-10	2022-

6 rows | 1-10 of 11 columns

```
summary(custom_prod)
```

```
##    id_prod              date                         session_id
##  Length:679332      Min.   :2021-03-01 00:01:07.83   Length:679332
##  Class :character   1st Qu.:2021-09-08 09:14:25.05   Class :character
##  Mode  :character   Median :2022-03-03 07:50:20.81   Mode  :character
##                     Mean   :2022-03-03 15:13:19.30
##                     3rd Qu.:2022-08-30 23:57:08.55
##                     Max.   :2023-02-28 23:58:30.78
##   client_id            price        categ        periode
##  Length:679332      Min.   :  0.62   0:415680   Length:679332
##  Class :character   1st Qu.:  8.87   1:227169   Class :character
##  Mode  :character   Median : 13.99   2: 36483   Mode  :character
##                     Mean   : 17.45
##                     3rd Qu.: 18.99
##                     Max.   :300.00
##   date_courte             sex                birth           age
##  Min.   :2021-03-01   Length:679332      Min.   :1929   Min.   :19.00
##  1st Qu.:2021-09-08   Class :character   1st Qu.:1970   1st Qu.:36.00
##  Median :2022-03-03   Mode  :character   Median :1980   Median :43.00
##  Mean   :2022-03-03                      Mean   :1978   Mean   :45.19
##  3rd Qu.:2022-08-30                      3rd Qu.:1987   3rd Qu.:53.00
##  Max.   :2023-02-28                      Max.   :2004   Max.   :94.00
```

Aucune valeur manquante.

Chiffre d'affaires par clients

```
CA_clients <- custom_prod %>%
  group_by(client_id) %>%
  summarise(CA = sum(price),
            prop = round(sum(price) / CA_total * 100, 2)) %>%
  arrange(desc(CA))
head(CA_clients, 10)
```

client_id <chr>	CA <dbl>	prop <dbl>
c_1609	324033.35	2.73
c_4958	289760.34	2.44
c_6714	153658.86	1.30
c_3454	113667.90	0.96
c_3263	5276.87	0.04
c_1570	5271.62	0.04
c_2899	5214.05	0.04
c_2140	5208.82	0.04
c_7319	5155.77	0.04

client_id <chr>	CA <dbl>	prop <dbl>
c_8026	5092.57	0.04

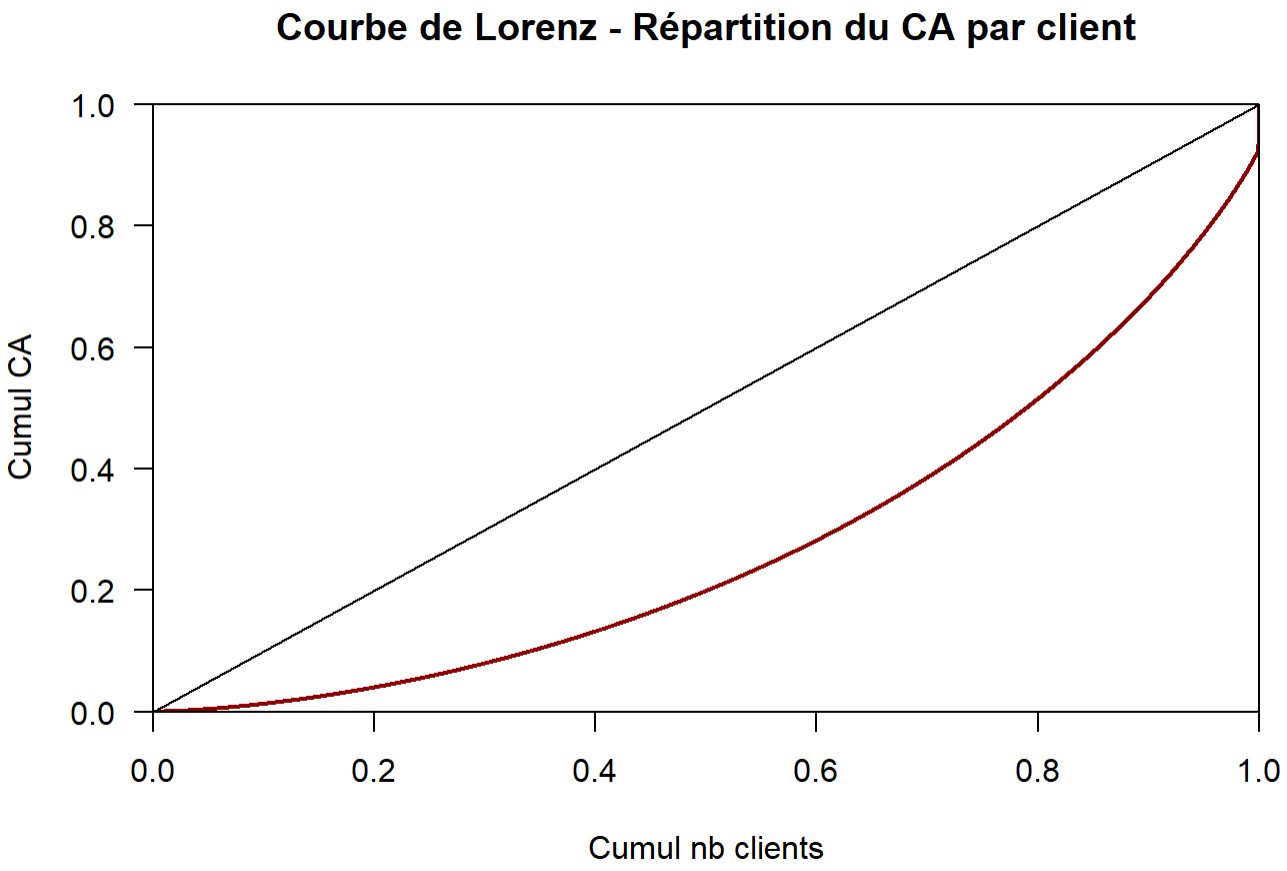
1-10 of 10 rows

```
# les 10 premiers clients par CA
```

4 gros clients (libraires) se détachent nettement et constituent, de fait, des outliers (CA > 100 k€ contre 5.3 k€ pour le 5e).

Mesure de l'inégalité de la répartition du chiffre d'affaires par client / Courbe de Lorenz

```
plot(Lc(CA_clients$CA), col = "darkred", lwd = 2, main="Courbe de Lorenz - Répartition du CA par client", xlab = "Cumul nb clients", ylab= "Cumul CA")
```



On compare les valeurs cumulées des déciles avec la droite d'équi-répartition, c'est une estimation de l'inégalité, ici moyennement marquée. Toutefois, une anomalie est repérée en fin de courbe. C'est l'influence des 4 clients libraires déjà signalés.

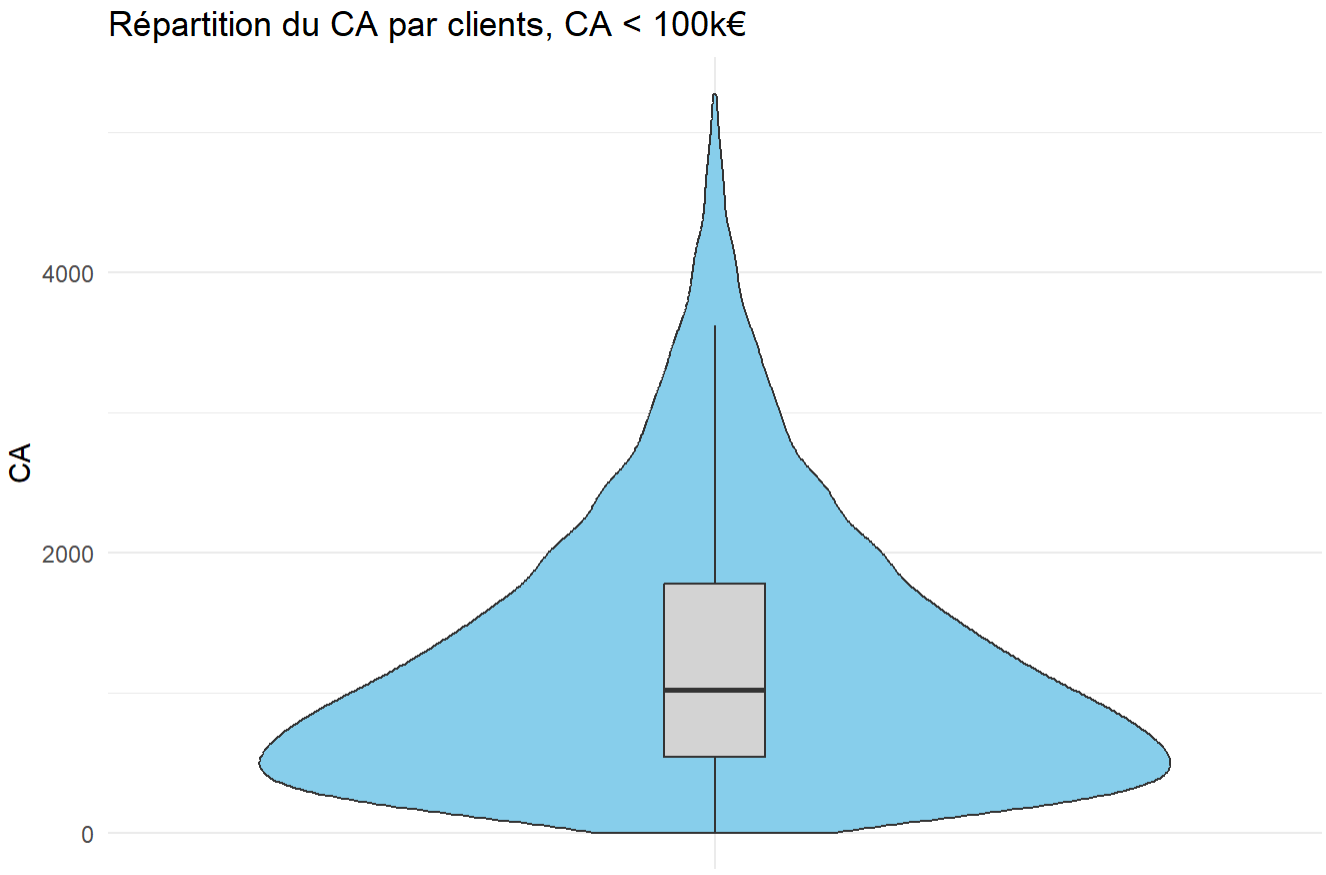
Indice ou coefficient de Gini

```
ineq(CA_clients$CA, type = "Gini") # G = 2 fois l'aire comprise entre la droite et la courbe / plus G est fort, plus l'inégalité est forte.
```

```
## [1] 0.4463949
```

Le coefficient de Gini est proche de 0.5, l'inégalité est moyennement marquée.

```
CA_clients %>%
  filter(CA_clients$CA < 100000) %>%
  ggplot(aes(x = "", y = CA))   +
  geom_violin(fill = "skyblue") +
  geom_boxplot(
    width = 0.1,
    outlier.shape = NA,
    show.legend = FALSE,
    fill = "lightgrey"
  ) +
  labs(title = "Répartition du CA par clients, CA < 100k€") +
  xlab("") +
  theme_minimal()
```



Une médiane aux alentours de 1200€ de CA, une évidente majorité de petits clients.

Etude des relations entre clientèle et produits

Existe-t-il une relation significative entre le genre de la clientèle et la catégorie de livres achetés ?

Tableau de contingence

```
sex_categ <- table(custom_prod$sex, custom_prod$categ)
sex_categ
```

```
##
##          0      1      2
##   f 206220 114899  17283
##   m 209460 112270  19200
```

Pearson's Chi-squared Test

```
chisq.test(sex_categ)
```

```
##
##  Pearson's Chi-squared test
##
## data:  sex_categ
## X-squared = 147, df = 2, p-value < 2.2e-16
```

Ici, la p-value est très proche de 0, en-dessous du seuil de décision choisi préalablement au test (5% retenu par défaut par le test du Chi2). On peut donc rejeter l'hypothèse d'indépendance des lignes et des colonnes du tableau. Sex et categ sont des variables corrélées.

```
chisq.residuals(sex_categ) # la fonction est attachée au package {questionr}
```

```
##
##         0     1     2
##   f -1.86  5.16 -6.61
##   m  1.85 -5.14  6.58
```

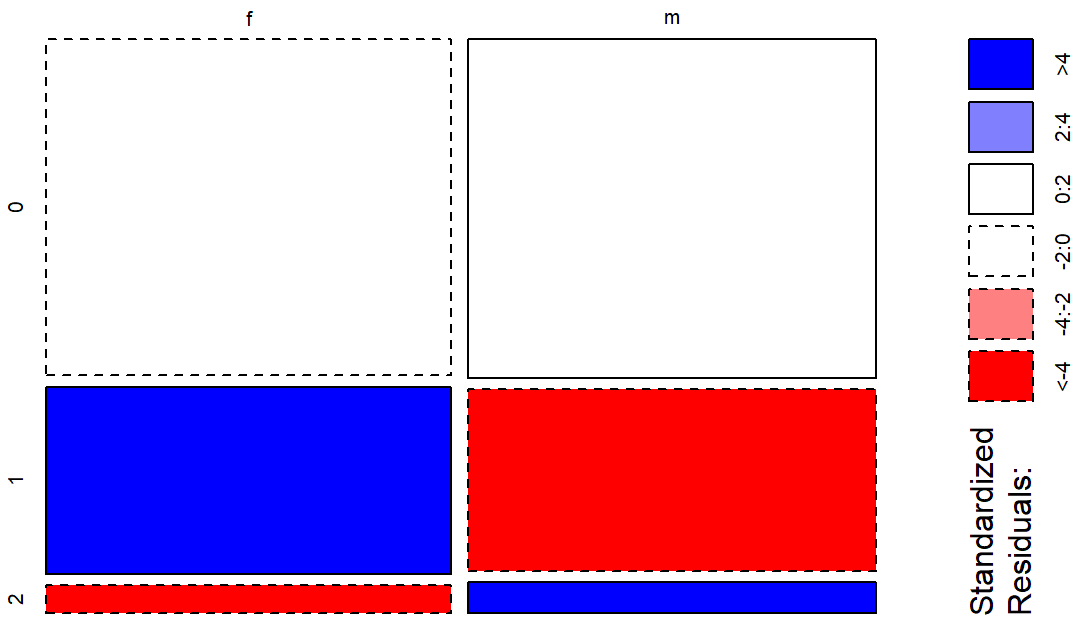
L'interprétation des résidus standardisés est la suivante :

- si la valeur du résidu pour une case est inférieure à -2, alors il y a une sous-représentation de cette case dans le tableau : les effectifs sont significativement plus faibles que ceux attendus sous l'hypothèse d'indépendance
- à l'inverse, si le résidu est supérieur à 2, il y a sur-représentation de cette case
- si le résidu est compris entre -2 et 2, il n'y a pas d'écart à l'indépendance significatif

Les indicateurs recueillis permettent de préciser : - que les femmes sont davantage attirées par les ouvrages de catégorie 1 que les hommes - que les hommes sont davantage attirés par les ouvrages de catégorie 2 que les femmes

C'est ce que permet de visualiser :

```
mosaicplot(sex_categ, shade = TRUE, main = "Analyse du tableau de contingence sexe ~ catégori
e")
```



En quoi l'âge des clients est-il corrélé aux variables : montant total des achats, fréquence d'achat, taille du panier moyen, catégories des livres achetés ?

Détermination des variables à considérer dans l'analyse

```
cumul_clients <- custom_prod %>%
  group_by(client_id) %>%
  summarise(
    CA = sum(price, na.rm = TRUE),
    freq = n_distinct(session_id),
    avg_basket = n() / freq,
    age = max(age)     # permet de conserver l'âge par-delà l'agrégation
  )

cumul_clients %>%
  arrange(desc(CA)) %>%
  head(4)
```

client_id	CA	freq	avg_basket	age
<chr>	<dbl>	<int>	<dbl>	<dbl>
c_1609	324033.3	10997	2.317723	43
c_4958	289760.3	3851	1.349000	24

client_id	CA	freq	avg_basket	age
<chr>	<dbl>	<int>	<dbl>	<dbl>
c_6714	153658.9	2620	3.506489	55
c_3454	113667.9	5573	1.215324	54

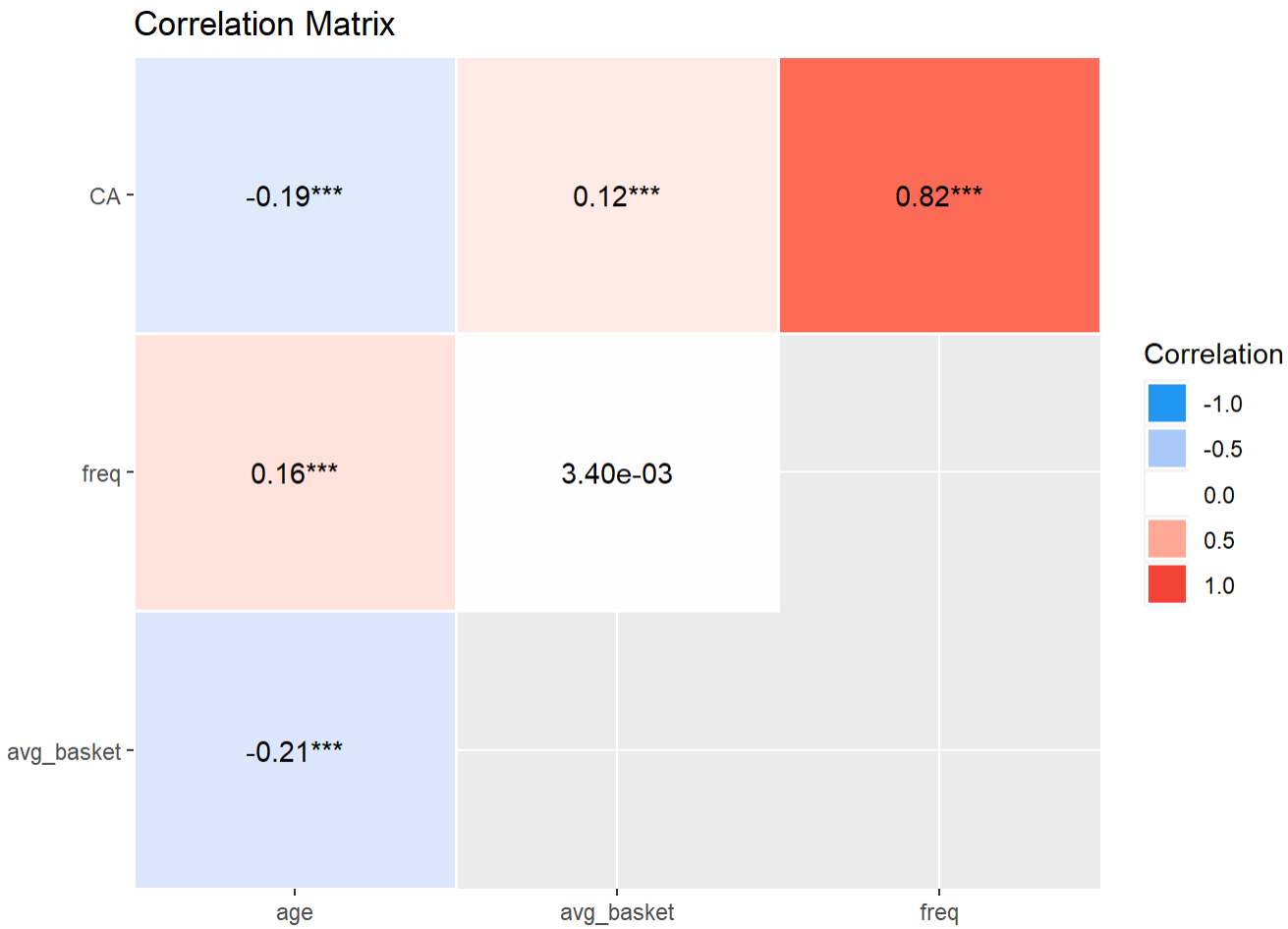
4 rows

Les 4 clients outliers (libraires) précédemment repérés ont une incidence très forte et relève d'un statut particulier. Ils sont à dissocier de l'analyse des relations entre les variables étudiées.

Matrice de corrélation - méthode de Pearson

```
# Suppression des outliers repérés
cumul_clients_wo_outliers <- cumul_clients %>%
   filter(CA < 100000)

library(correlation)
plot(summary(correlation(cumul_clients_wo_outliers)))
```



Les coefficients de corrélation en lien avec l'âge sont tous significatifs, certains corrélés négativement à l'âge, d'autres positivement.

Les régressions linéaires sont-elles un bon modèle de prédiction ?

Régressions linéaires portant sur les variables numériques

Sur l'âge et les données calculées

Age et chiffre d'affaires

```
reg_lin_age_CA <- lm(formula = CA ~ age, data = cumul_clients_wo_outliers)
summary(reg_lin_age_CA)
```

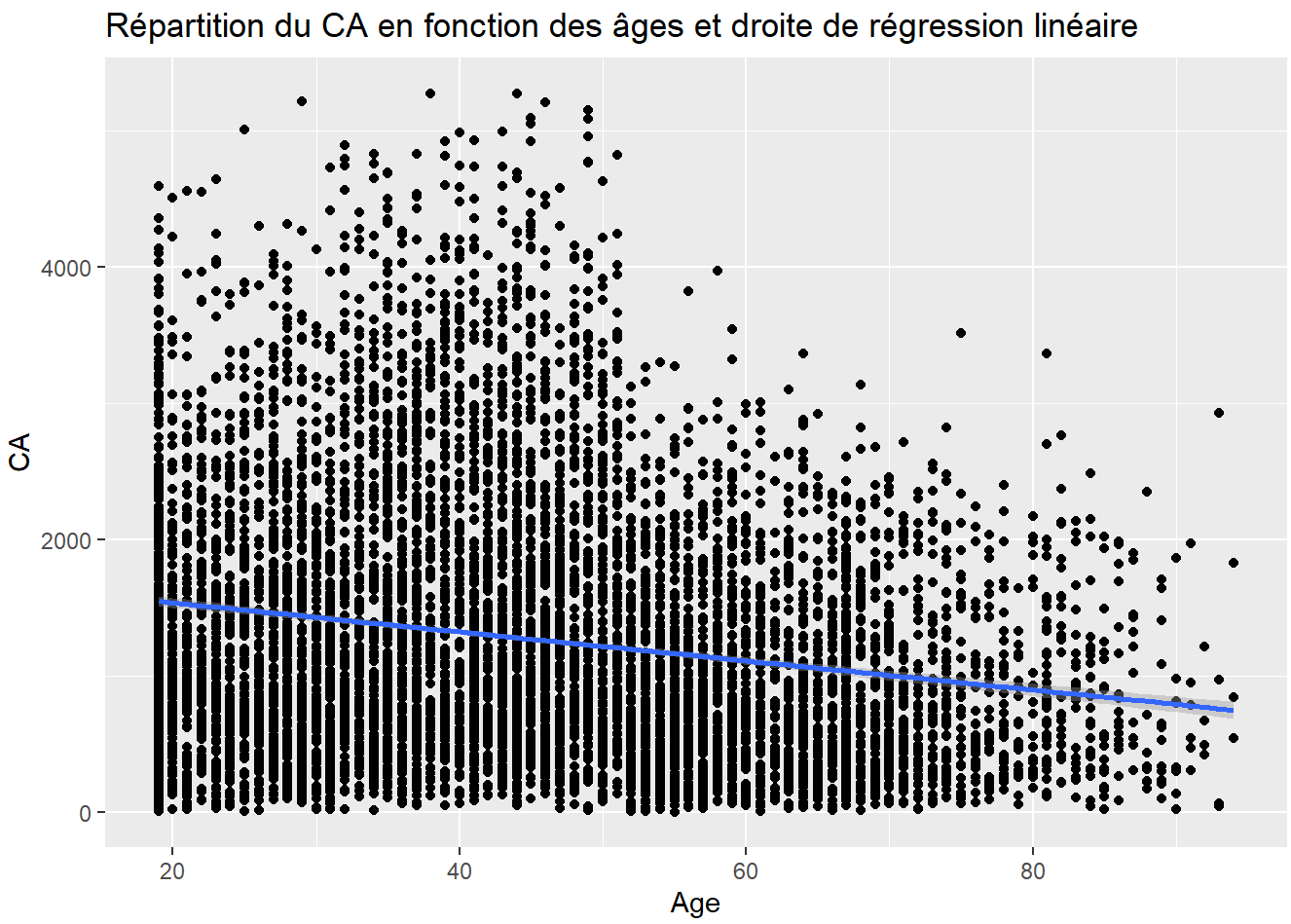
```
##
## Call:
## lm(formula = CA ~ age, data = cumul_clients_wo_outliers)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1539.1  -700.9  -235.9   500.1  3987.0
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1752.2781    28.5974   61.27   <2e-16 ***
## age          -10.6291     0.5979  -17.78   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 937.4 on 8594 degrees of freedom
## Multiple R-squared:  0.03547,    Adjusted R-squared:  0.03536
## F-statistic:   316 on 1 and 8594 DF,  p-value: < 2.2e-16
```

Au risque de 5%, l'hypothèse nulle est rejetée. Les variables sont corrélées, mais le coefficient de détermination R2 est très faible, le modèle linéaire représente imparfaitement le rapport entre les variables.

Ce que confirme la visualisation.

```
ggplot(cumul_clients_wo_outliers, aes(x = age, y = CA)) +
  geom_point() +
  stat_smooth(method = "lm") +
  xlab("Age") +
  ylab("CA")+
  ggtitle("Répartition du CA en fonction des âges et droite de régression linéaire")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

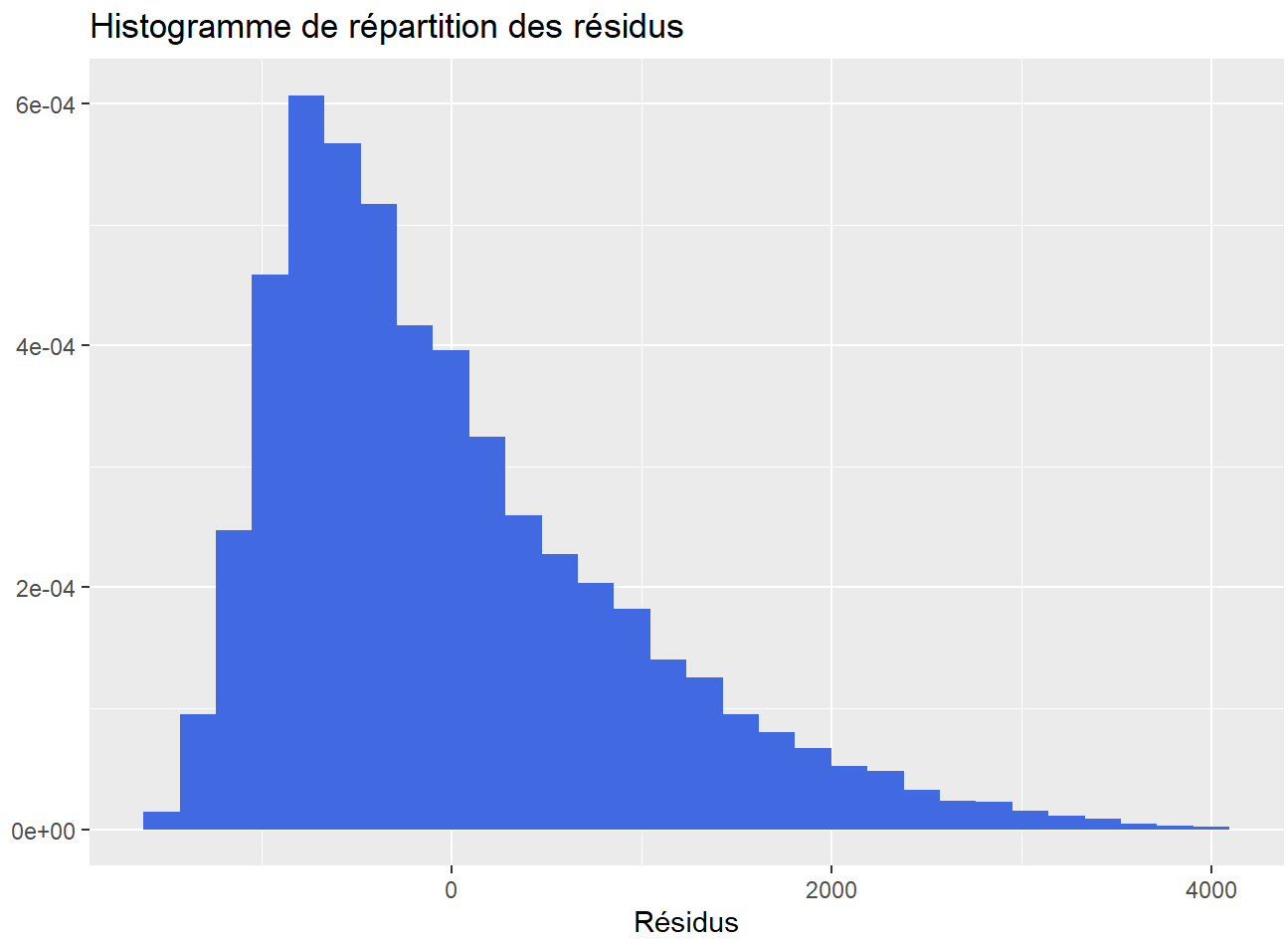


La pente est négative. Plus le client est âgé, moins il achète.

Vérification de la normalité des résidus

```
ggplot(cumul_clients_wo_outliers, aes(x = reg_lin_age_CA$residuals)) +
  geom_histogram(aes(y = after_stat(density)), fill = "royalblue") +
  ggtitle("Histogramme de répartition des résidus") +
  xlab("Résidus") +
  ylab("")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



On se rapproche très vaguement de la forme classique : centrée et symétrique.

Test de Kolmogorov-Smirnov

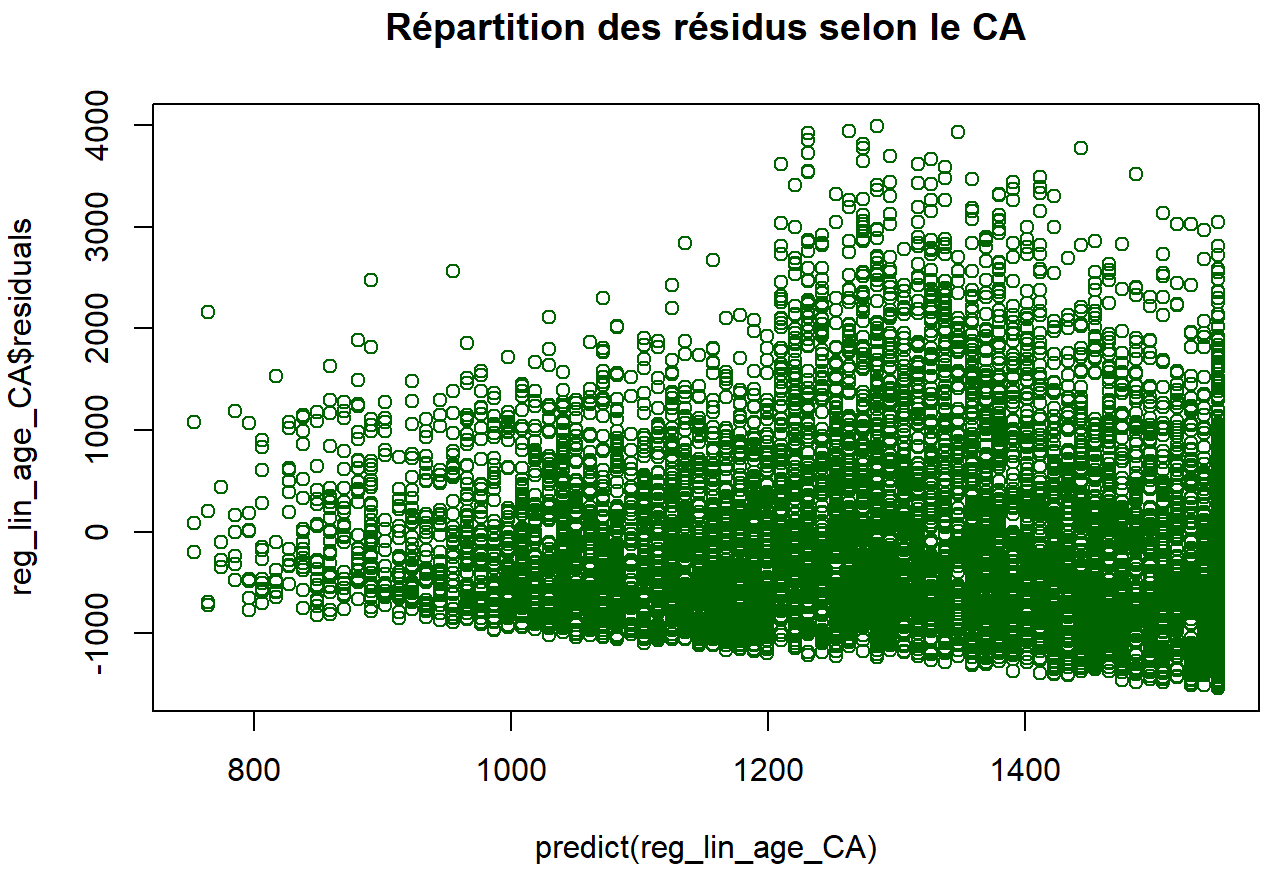
```
ks.test(reg_lin_age_CA$residuals, "pnorm")
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  reg_lin_age_CA$residuals
## D = 0.59523, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

Le test de normalité rejette l'hypothèse nulle : la distribution des résidus ne suit pas une loi normale.

Vérification de l'hétéroscédasticité (répartition inégale des résidus)

```
plot(predict(reg_lin_age_CA), reg_lin_age_CA$residuals, col="darkgreen", main="Répartition de
s résidus selon le CA")
```



La répartition des résidus est inégale, ce que confirme le **Test de Breusch-Pagan**

```
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
bptest(reg_lin_age_CA)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  reg_lin_age_CA
## BP = 142.62, df = 1, p-value < 2.2e-16
```

Au risque de 5% nous rejetons l'hypothèse nulle : il y a hétéroscédasticité. Pas de normalité des distributions, hétéroscedasticité des résidus, le modèle linéaire est peu assuré.

Age et moyenne du panier

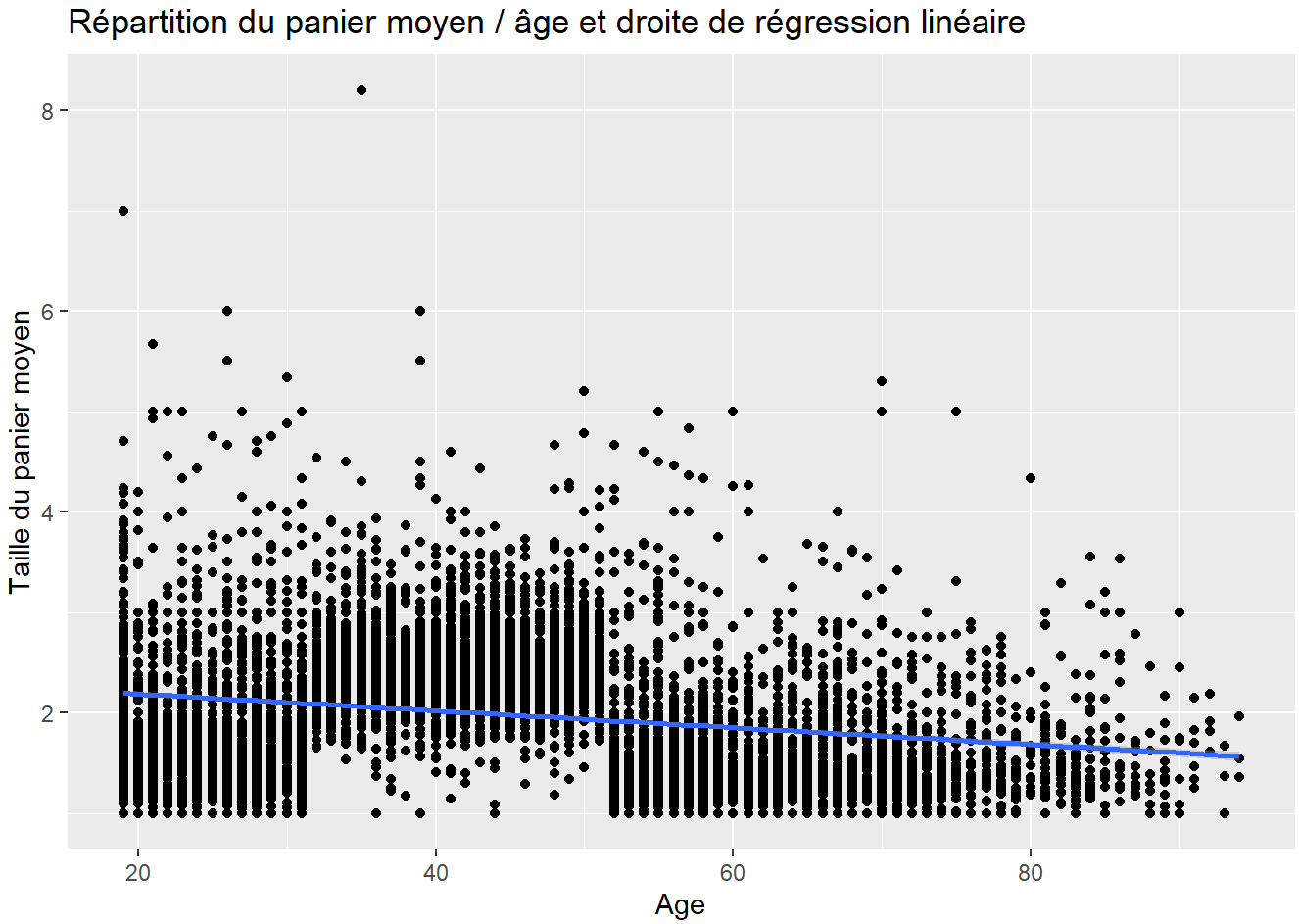
```
reg_lin_age_basket <- lm(formula =avg_basket ~ age, data = cumul_clients_wo_outliers)
summary(reg_lin_age_basket)
```

```
##
## Call:
## lm(formula = avg_basket ~ age, data = cumul_clients_wo_outliers)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1953 -0.5116 -0.0388  0.4184  6.1387
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.3544662  0.0198348   118.7   <2e-16 ***
## age         -0.0083750  0.0004147   -20.2   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6502 on 8594 degrees of freedom
## Multiple R-squared:  0.04531,    Adjusted R-squared:  0.0452
## F-statistic: 407.8 on 1 and 8594 DF,  p-value: < 2.2e-16
```

La p-value est proche de zéro, le R2 ajusté est faible, si les variables sont bien corrélées, la linéarité n'est pas le bon modèle. Ce que confirme la visualisation.

```
ggplot(cumul_clients_wo_outliers, aes(x = age, y = avg_basket)) +
  geom_point() +
  stat_smooth(method = "lm") +
  xlab("Age") +
  ylab("Taille du panier moyen") +
  ggtitle("Répartition du panier moyen / âge et droite de régression linéaire")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Tests de normalité de la distribution et d'hétéroscédasticité des résidus :

```
print(ks.test(reg_lin_age_basket$residuals, "pnorm"))
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  reg_lin_age_basket$residuals
## D = 0.14137, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

```
print(bptest(reg_lin_age_basket))
```

```
##
##  studentized Breusch-Pagan test
##
## data:  reg_lin_age_basket
## BP = 70.043, df = 1, p-value < 2.2e-16
```

Tout comme pour le CA, le modèle linéaire liant panier moyen à l'âge est mal assuré.

Age et fréquence d'achat

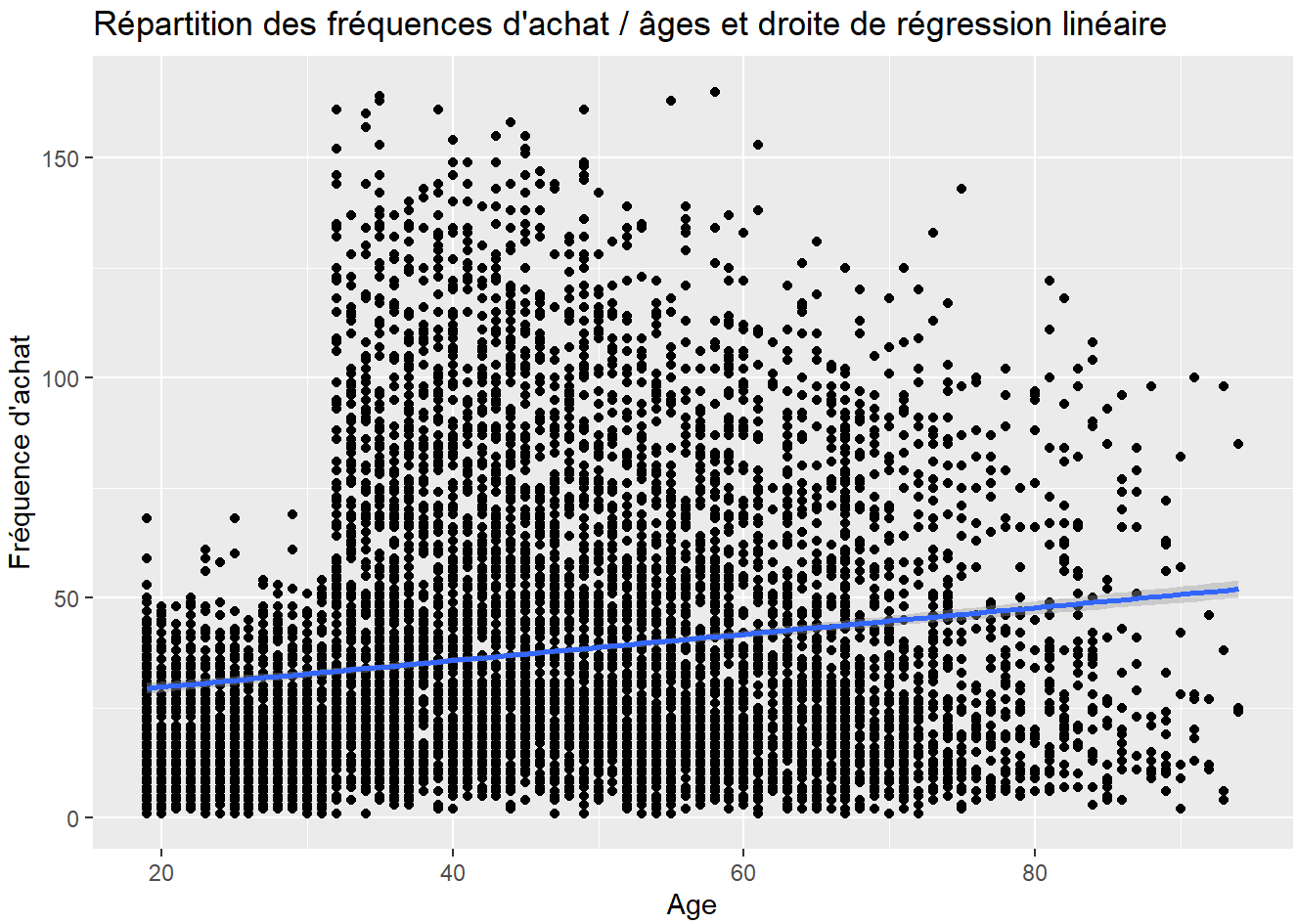
```
reg_lin_age_freq <- lm(formula = freq ~ age, data = cumul_clients_wo_outliers)
summary(reg_lin_age_freq)
```

```
##
## Call:
## lm(formula = freq ~ age, data = cumul_clients_wo_outliers)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -48.75 -21.41  -9.72  12.55 129.79
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 23.68989    0.92956   25.48   <2e-16 ***
## age          0.30068    0.01944   15.47   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 30.47 on 8594 degrees of freedom
## Multiple R-squared:  0.0271, Adjusted R-squared:  0.02698
## F-statistic: 239.3 on 1 and 8594 DF,  p-value: < 2.2e-16
```

La p-value permet de rejeter la nullité des coefficients. Le R2 est très faible. La linéarité explique peu la relation.

```
ggplot(cumul_clients_wo_outliers, aes(x = age, y = freq)) +
  geom_point() +
  stat_smooth(method = "lm") +
  xlab("Age") +
  ylab("Fréquence d'achat") +
  ggtitle("Répartition des fréquences d'achat / âges et droite de régression linéaire")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Tests de normalité de la distribution et d'hétéroscédasticité des résidus :

```
print(ks.test(reg_lin_age_freq$residuals, "pnorm"))
```

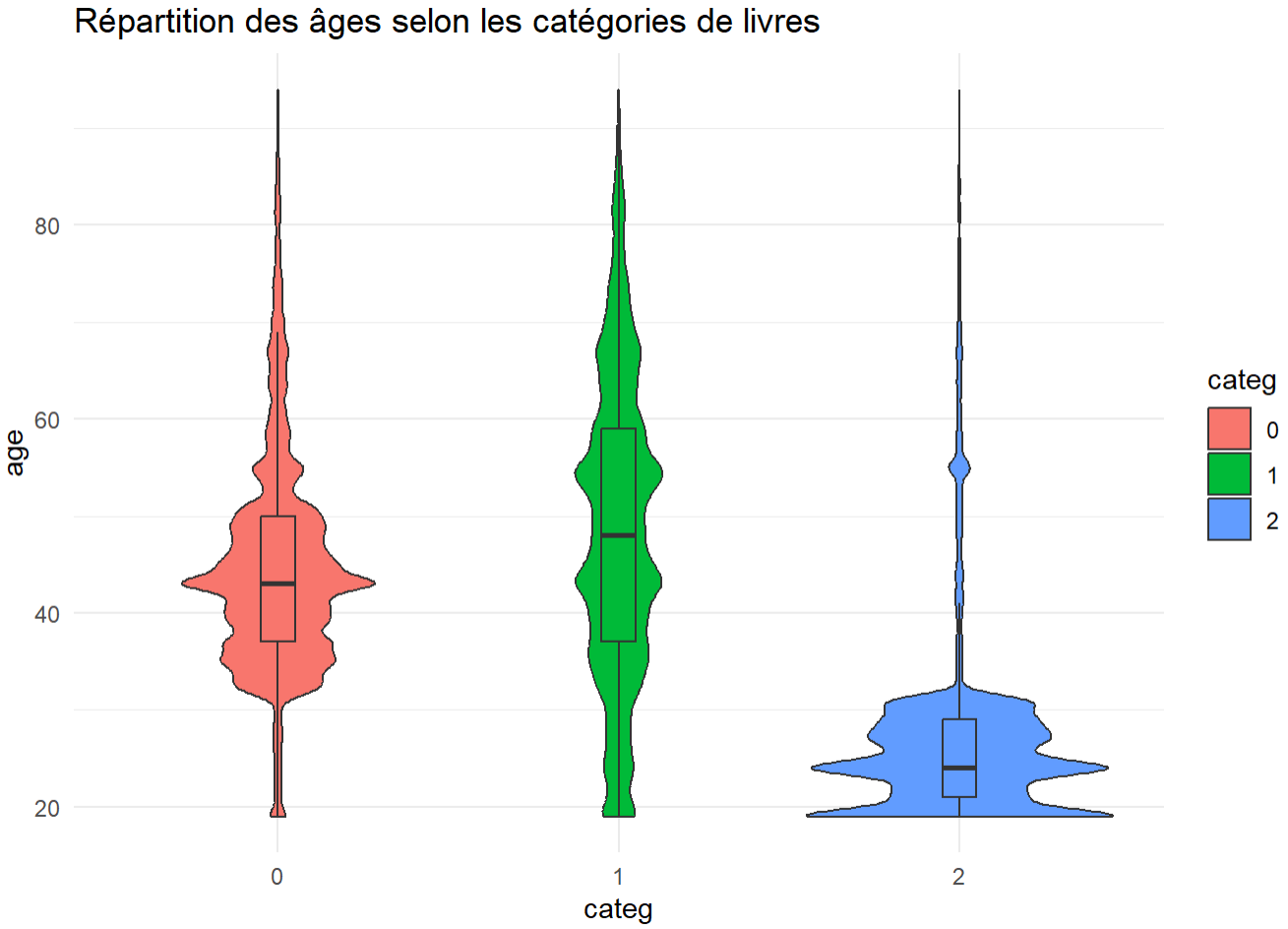
```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  reg_lin_age_freq$residuals
## D = 0.59697, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

```
print(bptest(reg_lin_age_freq))
```

```
##
##  studentized Breusch-Pagan test
##
## data:  reg_lin_age_freq
## BP = 61.355, df = 1, p-value = 4.765e-15
```

Relations entre les âges et les catégories

```
ggplot(custom_prod) +
  aes(x = categ, y = age, fill = categ) +
  geom_violin(adjust = 1L,
              scale = "area") +
  geom_boxplot(width = 0.1,
               outlier.shape = NA,
               show.legend = FALSE) +
  theme_minimal() +
  ggtitle("Répartition des âges selon les catégories de livres")
```



Il existe une tendance forte des plus jeunes sur la catégorie 2. On constate beaucoup de valeurs extrêmes et d'amplitude pour chacune des catégories. Les conditions sont-elles réunies pour réaliser une analyse des variances ?

```
res_aov <- aov(age ~ categ,
               data = custom_prod)

par(mfrow = c(1, 2)) # combine plots

# histogram
hist(res_aov$residuals, main ="Histogramme des résidus")

# QQ-plot
library(car)
```

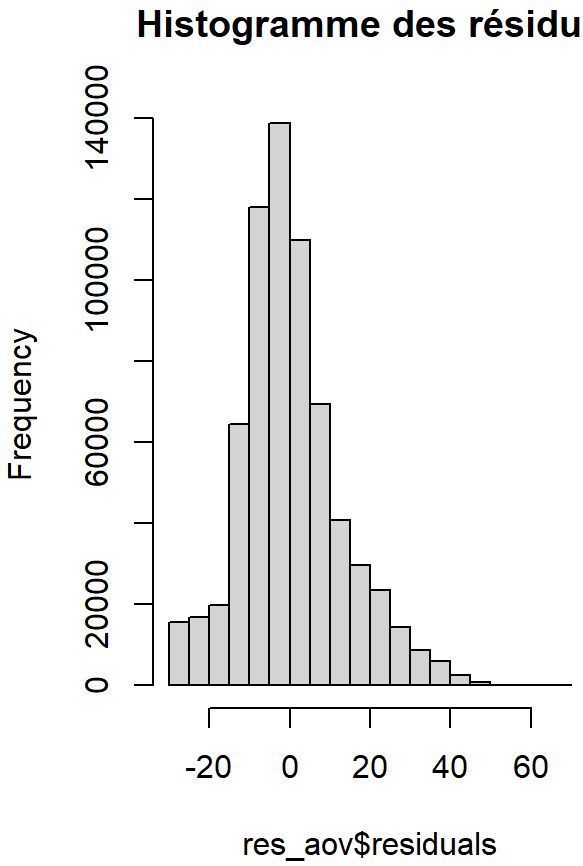
```
## Loading required package: carData
```

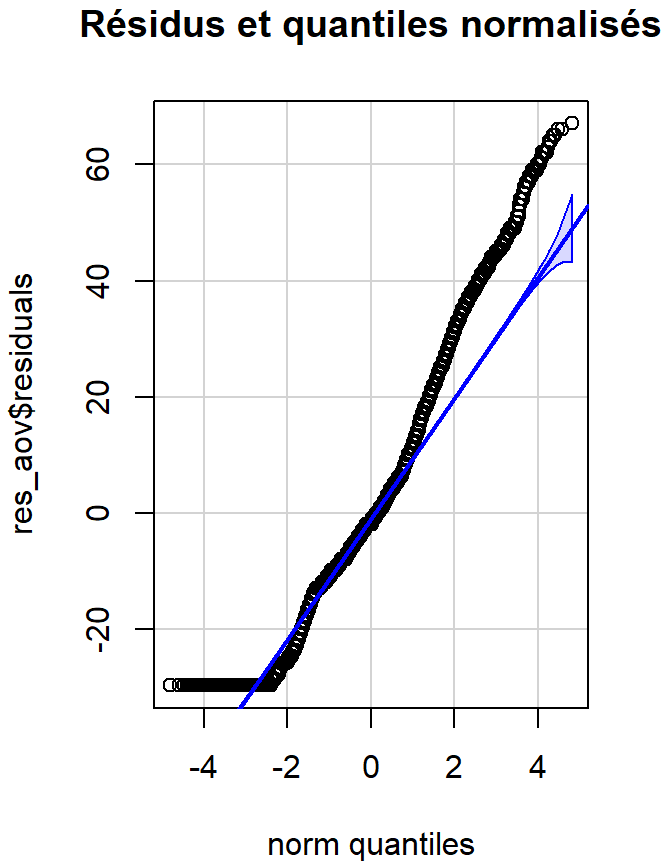
```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
## The following object is masked from 'package:purrr':
##
##     some
```

```
qqPlot(res_aov$residuals, main = "Résidus et quantiles normalisés",
  id = FALSE # id = FALSE to remove point identification
)
```





Quantile-Comparison Plot : les résidus décrochent de la droite de référence. On s'écarte sensiblement de la loi normale.

Test de normalité de distribution des échantillons (Kolmogorov-Smirnov Test)

```
print(ks.test(custom_prod[custom_prod$categ=="0",]$age, "pnorm"))
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  custom_prod[custom_prod$categ == "0", ]$age
## D = 1, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

```
print(ks.test(custom_prod[custom_prod$categ=="1",]$age, "pnorm"))
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  custom_prod[custom_prod$categ == "1", ]$age
## D = 1, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

```
print(ks.test(custom_prod[custom_prod$categ=="2",]$age, "pnorm"))
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  custom_prod[custom_prod$categ == "2", ]$age
## D = 1, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

Les tests permettent de rejeter la normalité de distribution des 3 catégories.

```
# Levene's test
leveneTest(age ~ categ,
               data = custom_prod
)
```

	Df <int>	F value <dbl>	Pr(>F) <dbl>
group	2	26091.73	0
	679329	*NA*	*NA*

2 rows

Les distributions ne suivent pas une loi normale, pas d'homogénéité des variances, les tests d'analyse paramétrique de la variance Anova ou Welch Anova ne sont pas adaptés.

Recours à un test non paramétrique : test de Kruskal-Wallis.

```
kruskal.test(custom_prod$age, custom_prod$categ) # test non paramétrique
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  custom_prod$age and custom_prod$categ
## Kruskal-Wallis chi-squared = 79351, df = 2, p-value < 2.2e-16
```

Au risque de 5%, nous pouvons rejeter l'hypothèse d'indépendance des variables, elle sont corrélées.

```
custom_prod %>%
  group_by(categ) %>%
  summarise(
    age_max = max(age),
    age_min = min(age),
    age_mean = mean(age),
    age_median = median(age),
    age_sd = sd(age),
    count = n()
  )
```

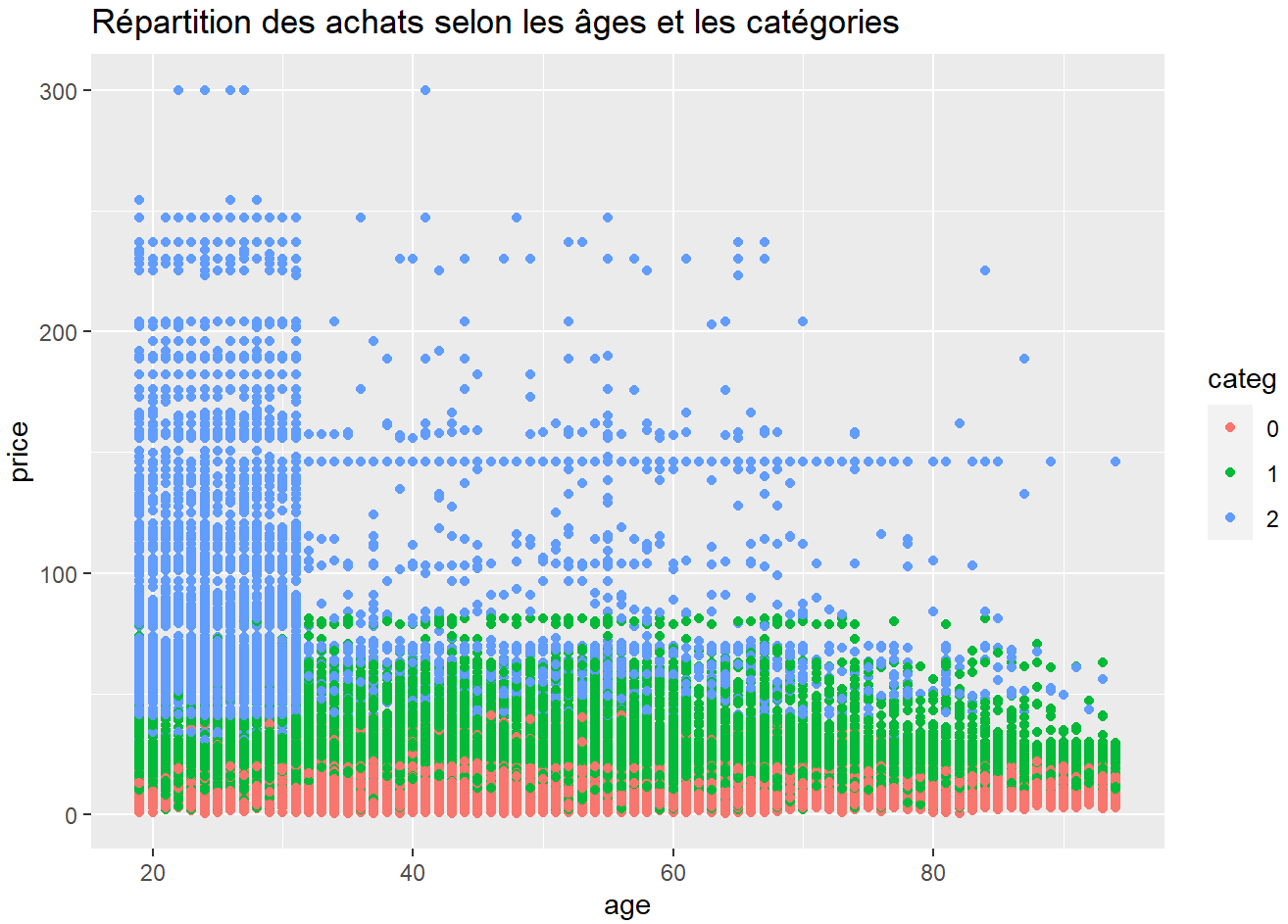
categ <fct>	age_max <dbl>	age_min <dbl>	age_mean <dbl>	age_median <dbl>	age_sd <dbl>	count <int>
0	94	19	44.89762	43	11.209482	415680
1	94	19	48.65124	48	15.495910	227169
2	94	19	26.94803	24	9.798143	36483

3 rows

La catégorie 2 se détache nettement par son public sensiblement plus jeune. Les moyennes d'âges sont similaires entre les catégories 0 et 1. Les mêmes âges extrêmes sont présents sur les 3 catégories.

Approche visuelle affinée des différences d'achats de livres selon les âges :

```
ggplot(custom_prod) +
  geom_point(
    aes(x = age, y = price, color = categ)
  ) +
  ggtitle("Répartition des achats selon les âges et les catégories")
```


Répartition des achats selon les âges et les catégories
price
age
categ
0
1
2
300
200
100
0
20
40
60
80