Analyze the sales of a bookstore

Corentin Casali

2/1/2022

Data Analyst Consultant at Lapage, a large and reputable generalist online bookstore. You will report directly to the Marketing department. Lapage was originally a physical bookstore with several points of sale. However, due to the success of some of its products and the enthusiasm of its customers, it decided to open an online sales site two years ago. You intervene because the company wishes to take stock after two years of exercise, and to be able to analyze its strong points, its weak points, the customers behaviors, etc…

Librairies - R.packages

Import of R packages useful for the analysis.

```
## First specify the packages of interest
packages = c("knitr","readr","dplyr","kableExtra",'stringi','tidyverse','lubridate','moments','scales','zoo',"gri
dExtra","plotly","rstatix","ggpubr",'rstudioapi','ineq')

## Now load or install&load all
package.check <- lapply(
  packages,
  FUN = function(x) {
    if (!require(x, character.only = TRUE)) {
      install.packages(x, dependencies = TRUE)
      library(x, character.only = TRUE)
    }
  }
)
# setwd(dirname(getActiveDocumentContext()$path))       # Set working directory to source file location
knitr::opts_chunk$set(echo=TRUE, fig.height = 8, fig.width = 12, fig.align = "center")
```

Data cleaning :

Import of data into 3 different dataframes

```
# Loading .csv files into dataframes
df_customers <- read_csv("data/customers.csv")
df_products <- read_csv("data/products.csv")
df_transactions <-  read_csv("data/transactions.csv")
```

Quick view of our dataframes:

```
# Visualization of dataframes
str(df_customers)
```

```
## spec_tbl_df [8,623 × 3] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ client_id: chr [1:8623] "c_4410" "c_7839" "c_1699" "c_5961" ...
##  $ sex      : chr [1:8623] "f" "f" "f" "f" ...
##  $ birth    : num [1:8623] 1967 1975 1984 1962 1943 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   client_id = col_character(),
##   ..   sex = col_character(),
##   ..   birth = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
str(df_products)
```

```
## spec_tbl_df [3,287 × 3] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ id_prod: chr [1:3287] "0_1421" "0_1368" "0_731" "1_587" ...
##  $ price  : num [1:3287] 19.99 5.13 17.99 4.99 3.99 ...
##  $ categ  : num [1:3287] 0 0 0 1 0 0 1 0 0 0 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   id_prod = col_character(),
##   ..   price = col_double(),
##   ..   categ = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
str(df_transactions)
```

```
## spec_tbl_df [679,532 × 4] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ id_prod   : chr [1:679532] "0_1518" "1_251" "0_1277" "2_209" ...
##  $ date      : POSIXct[1:679532], format: "2022-05-20 13:21:29" "2022-02-02 07:55:19" ...
##  $ session_id: chr [1:679532] "s_211425" "s_158752" "s_225667" "s_52962" ...
##  $ client_id : chr [1:679532] "c_103" "c_8534" "c_6714" "c_6941" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   id_prod = col_character(),
##   ..   date = col_datetime(format = ""),
##   ..   session_id = col_character(),
##   ..   client_id = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

Missing values

We check if our data have missing values.

```
# Handling missing values
sapply(df_customers, function(y) sum(is.na(y)))
```

```
## client_id       sex     birth
##         0         0         0
```

```
sapply(df_products, function(y) sum(is.na(y)))
```

```
## id_prod   price   categ
##       0       0       0
```

```
sapply(df_transactions, function(y) sum(is.na(y)))
```

```
##    id_prod       date session_id  client_id
##          0        200          0          0
```

The **transactions** dataframe has ***200 missing values*** on the ***date*** variable.

```
# Viewing missing data:
kable(head(df_transactions[is.na(df_transactions$date),],10))%>%
  kable_styling(latex_options = 'striped')
```

id_prod	date	session_id	client_id
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

```
# Delete NA values from the "date" variable:
df_transactions <- df_transactions %>% drop_na(date)

# Check :
sapply(df_transactions, function(y) sum(is.na(y)))
```

```
##    id_prod       date session_id  client_id
##          0          0          0          0
```

We have removed the NA data from the date variable. Indeed, we found that it is probably a test variable. They will not be necessary in our data analysis.

Duplicates :

We check if our data have duplicates.

```
sum(duplicated(df_customers))
```

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

```
sum(duplicated(df_products))
```

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

```
sum(duplicated(df_transactions))
```

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

No duplicates in our dataframes!

Specific inspection of dataframes

Dataframe - customers :

```
# Dataframe inspection:
kable(head(df_customers)) %>%
  kable_styling(latex_options = 'striped')
```

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

```
# Check ID :
kable(head(df_customers[order(df_customers$client_id, decreasing=TRUE),])) %>%
  kable_styling(latex_options = 'striped')
```

client_id	sex	birth
ct_1	m	2001
ct_0	f	2001
c_999	m	1964
c_998	m	2001
c_997	f	1994
c_996	f	1970

```
kable(head(df_customers[order(df_customers$client_id, decreasing=FALSE),])) %>%
  kable_styling(latex_options = 'striped')
```

client_id	sex	birth
c_1	m	1955
c_10	m	1956
c_100	m	1992
c_1000	f	1966
c_1001	m	1982

c_1002 f 1982

```
# Check "âges" :
summary(df_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
```

```
# List of IDs to check
checkID <- c('ct_0','ct_1')
```

We notice several things in this dataframe:

- the **ct_0** and **ct_1** IDs do not have the same mapping as the other IDs They could potentially be **test id**". We will see later what their transactions correspond to once the joins are done.
- the ages of the clients range from 19 to 94. Therefore, we do not have any anomalies regarding the ages.

Dataframe - products :

```
# Dataframe inspection:
kable(head(df_products)) %>%
  kable_styling(latex_options = 'striped')
```

id_prod	price	categ
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

```
# Checking outliers:
summary(df_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
```

We notice negative prices. So we will do a cleaning of the data of the variable **price**.

```
# Data whose price is < 0
kable(df_products[df_products$price <= 0,]) %>%
   kable_styling(latex_options = 'striped')
```

id_prod	price	categ
T_0	-1	0

```
# Deleting this data
df_products <- df_products[df_products$price >= 0,]

# Checking the cleaning
summary(df_products$price)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.62    6.99   13.07   21.86   22.99  300.00
```

So we have prices ranging from 0,62€ to 300€. No more price anomalies.

Dataframe - transaction :

```
kable(head(df_transactions)) %>%
  kable_styling(latex_options = 'striped')
```

id_prod	date	session_id	client_id
0_1518	2022-05-20 13:21:29	s_211425	c_103
1_251	2022-02-02 07:55:19	s_158752	c_8534
0_1277	2022-06-18 15:44:33	s_225667	c_6714
2_209	2021-06-24 04:19:29	s_52962	c_6941
0_1509	2023-01-11 08:22:08	s_325227	c_4232
0_1418	2022-10-20 15:59:16	s_285425	c_1478

```
summary(df_transactions)
```

```
##    id_prod              date                       session_id
##  Length:679332      Min.   :2021-03-01 00:01:07   Length:679332
##  Class :character   1st Qu.:2021-09-08 09:14:25   Class :character
##  Mode  :character   Median :2022-03-03 07:50:20   Mode  :character
##                     Mean   :2022-03-03 15:13:19
##                     3rd Qu.:2022-08-30 23:57:08
##                     Max.   :2023-02-28 23:58:30
##   client_id
##  Length:679332
##  Class :character
##  Mode  :character
##
##
##
```

No data to clean up for the transaction dataframe.

Data join :

Summary of the 3 dataframes:

Qualitative variables:

- id (id_prod, client_id, session_id);
- gender of clients (f or m) ;
- product categories (0, 1 or 2) ;
- transaction dates

Quantitative variable:

- product price

Join between the transactions and customers dataframe

```
# Join between the transactions and customers dataframe
df_transactions_customers <- left_join(df_transactions, df_customers, by="client_id")
kable(head(df_transactions_customers)) %>%
  kable_styling(latex_options = 'striped')
```

id_prod	date	session_id	client_id	sex	birth
0_1518	2022-05-20 13:21:29	s_211425	c_103	f	1986
1_251	2022-02-02 07:55:19	s_158752	c_8534	m	1988
0_1277	2022-06-18 15:44:33	s_225667	c_6714	f	1968
2_209	2021-06-24 04:19:29	s_52962	c_6941	m	2000
0_1509	2023-01-11 08:22:08	s_325227	c_4232	m	1980
0_1418	2022-10-20 15:59:16	s_285425	c_1478	f	1972

```
# Checking missing values :
sapply(df_transactions_customers, function(y) sum(is.na(y)))
```

```
##    id_prod       date session_id  client_id        sex      birth
##          0          0          0          0          0          0
```

```
#Dataframe size
nrow(df_transactions)
```

```
## [1] 679332
```

```
nrow(df_transactions_customers)
```

```
## [1] 679332
```

No missing data. Also, we have the same number of rows, so the join worked.

Final dataframe join

```
df_final <- left_join(df_transactions_customers,df_products, by="id_prod")
kable(head(df_final))%>%
  kable_styling(latex_options = 'striped')
```

id_prod	date	session_id	client_id	sex	birth	price	categ
0_1518	2022-05-20 13:21:29	s_211425	c_103	f	1986	4.18	0
1_251	2022-02-02 07:55:19	s_158752	c_8534	m	1988	15.99	1
0_1277	2022-06-18 15:44:33	s_225667	c_6714	f	1968	7.99	0
2_209	2021-06-24 04:19:29	s_52962	c_6941	m	2000	69.99	2
0_1509	2023-01-11 08:22:08	s_325227	c_4232	m	1980	4.99	0
0_1418	2022-10-20 15:59:16	s_285425	c_1478	f	1972	8.57	0

```
# Checking missing values :
sapply(df_final, function(y) sum(is.na(y)))
```

```
##    id_prod       date session_id  client_id        sex      birth      price
##          0          0          0          0          0          0        221
##      categ
##        221
```

221 missing values for the **price** variable & the **categ** variable.

```
# Creation of a dataframe containing only the NA of the *price* variable of the final dataframe
df_no_values <- df_final[is.na(df_final$price)== TRUE,]
kable(head(df_no_values))%>%
  kable_styling(latex_options = 'striped')
```

id_prod	date	session_id	client_id	sex	birth	price	categ
0_2245	2022-09-23 07:22:38	s_272266	c_4746	m	1940	NA	NA
0_2245	2022-07-23 09:24:14	s_242482	c_6713	f	1963	NA	NA
0_2245	2022-12-03 03:26:35	s_306338	c_5108	m	1978	NA	NA
0_2245	2021-08-16 11:33:25	s_76493	c_1391	m	1991	NA	NA
0_2245	2022-07-16 05:53:01	s_239078	c_7954	m	1973	NA	NA
0_2245	2023-01-21 18:39:25	s_330241	c_6268	m	1991	NA	NA

```
sapply(df_no_values, function(y) sum(is.na(y)))
```

```
##    id_prod       date session_id  client_id        sex      birth      price
##          0          0          0          0          0          0        221
##      categ
##        221
```

The variables **price** and **categ** are correlated for missing data.

```
# Product inspection associated with these variables
unique(df_no_values$id_prod)
```

```
## [1] "0_2245"
```

```
# Product inspection
kable(head(df_final[df_final$id_prod=='0_2245',]))%>%
  kable_styling(latex_options = 'striped')
```

id_prod	date	session_id	client_id	sex	birth	price	categ
0_2245	2022-09-23 07:22:38	s_272266	c_4746	m	1940	NA	NA
0_2245	2022-07-23 09:24:14	s_242482	c_6713	f	1963	NA	NA
0_2245	2022-12-03 03:26:35	s_306338	c_5108	m	1978	NA	NA
0_2245	2021-08-16 11:33:25	s_76493	c_1391	m	1991	NA	NA
0_2245	2022-07-16 05:53:01	s_239078	c_7954	m	1973	NA	NA
0_2245	2023-01-21 18:39:25	s_330241	c_6268	m	1991	NA	NA

```
# Number of transactions
nrow(df_final[df_final$id_prod=='0_2245',])
```

```
## [1] 221
```

We realize that only one product is associated with these missing data. It is the product : **0_2245**. It is indeed a product that does not contain **price** and **categ**. However, the customers are different. We can assume that it is a product offered or a coupon issued by the company that does not belong to any category and therefore has no price. We will therefore remove this **id_prod** from the final dataframe.

```
# Delete the product id 0_2245
df_final <- subset(df_final,id_prod!='0_2245')

# Adding a date variable and separating it into year, day, month for processing
df_final$newDate <- date(df_final$date)
df_final$year <- year(df_final$newDate)
df_final$month <- month(df_final$newDate)
df_final$day <- day(df_final$newDate)

# Factor in the category
df_final$categ <- as.factor(df_final$categ)
```

To facilitate future processing, we have segmented the date of transactions as well as factoring the product categories. As a result, our final dataframe looks like this:

```
kable(head(df_final))%>%
  kable_styling(latex_options = 'striped')
```

id_prod	date	session_id	client_id	sex	birth	price	categ	newDate	year	month	day
0_1518	2022-05-20 13:21:29	s_211425	c_103	f	1986	4.18	0	2022-05-20	2022	5	20
1_251	2022-02-02 07:55:19	s_158752	c_8534	m	1988	15.99	1	2022-02-02	2022	2	2
0_1277	2022-06-18 15:44:33	s_225667	c_6714	f	1968	7.99	0	2022-06-18	2022	6	18
2_209	2021-06-24 04:19:29	s_52962	c_6941	m	2000	69.99	2	2021-06-24	2021	6	24
0_1509	2023-01-11 08:22:08	s_325227	c_4232	m	1980	4.99	0	2023-01-11	2023	1	11
0_1418	2022-10-20 15:59:16	s_285425	c_1478	f	1972	8.57	0	2022-10-20	2022	10	20

```
# Vérification des ID test :
subset(df_final, client_id %in% checkID)
```

```
## # A tibble: 0 × 12
## # … with 12 variables: id_prod <chr>, date <dttm>, session_id <chr>,
## #   client_id <chr>, sex <chr>, birth <dbl>, price <dbl>, categ <fct>,
## #   newDate <date>, year <dbl>, month <dbl>, day <int>
```

We get no data, the client_id **ct_0** and **ct_1** were indeed test IDs. Their transactions have been deleted.

Antoine’s mission:

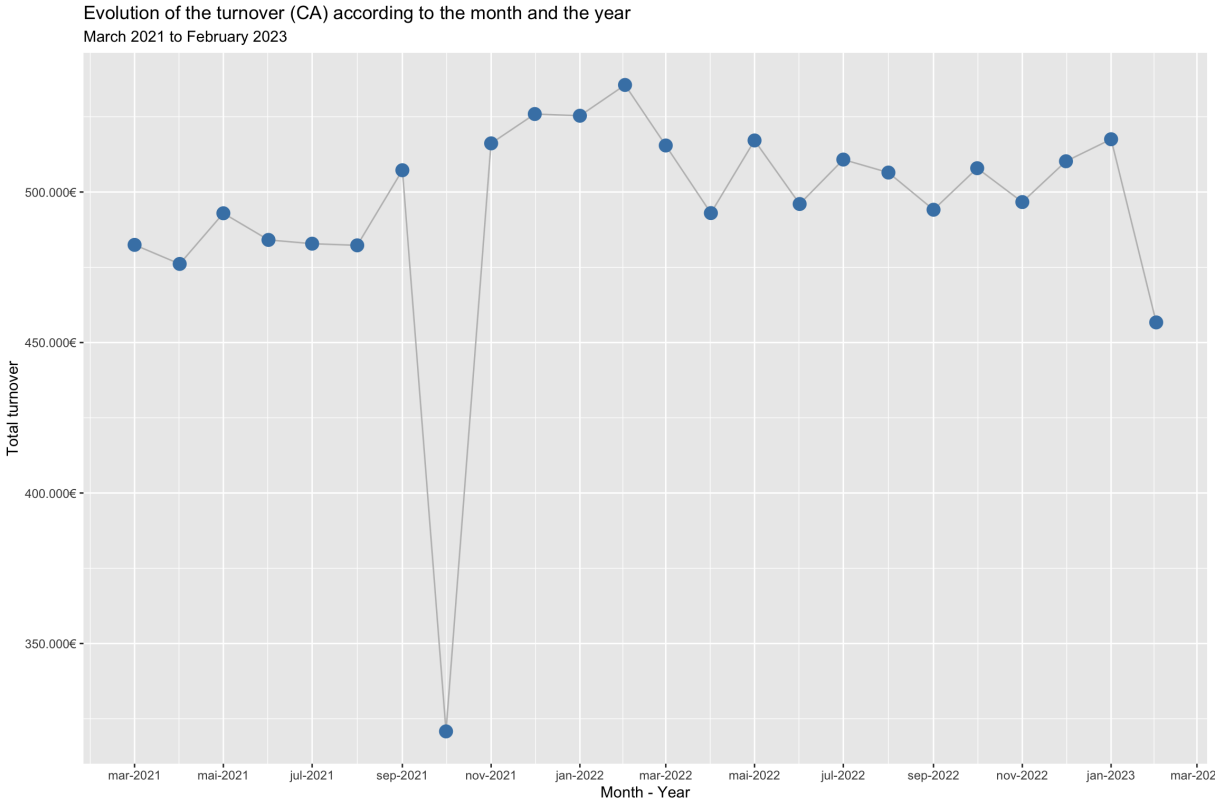
- Indicators and graphs around the turnover;
- Decomposition of the turnover in moving average to evaluate the global trend;
- Zoom on references: top/flop; distribution by category
- Information on customer profiles: breakdown of sales, Lorenz curve.

Analysis and evolution of the turnover

```
# Creation of a dataframe recovering for each month and year the turnover
monthsCA <- df_final %>%
  mutate(month = sprintf("%02d",df_final$month))%>%
  group_by(year, month) %>%
  summarise(totalCAk = sum(price, na.rm=T))
```

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

```
# Add monthly date (Date format)
monthsCA$yearMonth <- paste(monthsCA$year,monthsCA$month,'1',sep='-')
monthsCA$date <- as.Date(monthsCA$yearMonth, "%Y-%m-%d")

monthsCA %>%
  ggplot(aes(x=date,y=totalCAk,group=1))+
  geom_line(color="grey")+
  geom_point(color='steelblue',size=4) +
  scale_x_date(date_labels = ('%b-%Y'), date_breaks= "2 month")+
  scale_y_continuous(labels = dollar_format(suffix = "€", prefix="", big.mark = '.', decimal.mark = ','))+
  labs(title="Evolution of the turnover (CA) according to the month and the year",
       subtitle = "March 2021 to February 2023",
       x="Month - Year",
       y="Total turnover")
```



There is a drop in turnover in October 2021. Nevertheless, we are going to make a moving average to see the global trends, as well as a breakdown of the turnover in week and not in month.

Moving average of the turnover / Global trend :

```
# Breakdown of the df_final into year, month, week, category to make a detailed analysis.
weekCA <- df_final %>%
  mutate(month = sprintf("%02d",month),
         week = sprintf("%02d",week(newDate)),
         idDate = paste(year,month,week,sep='')) %>%
  group_by(year,month, week, categ, idDate) %>%
  summarise(CA = sum(price, na.rm=T))
```

```
## `summarise()` has grouped output by 'year', 'month', 'week', 'categ'. You can override using the `.groups` arg
ument.
```

```
# We transform week 53 into 52 so as not to overestimate the cutting
weekCA$week[weekCA$week==53] <- 52

# Breakdown of the df_final in year, month, week to make the moving average of the turnover
globalCA <- df_final %>%
  mutate(month = sprintf("%02d",month),
         week = sprintf("%02d",week(newDate)),
         idDate = paste(year,month,week,sep='')) %>%
  group_by(year,month, week, idDate) %>%
  summarise(totalCA = sum(price, na.rm=T)) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'year', 'month', 'week'. You can override using the `.groups` argument.
```

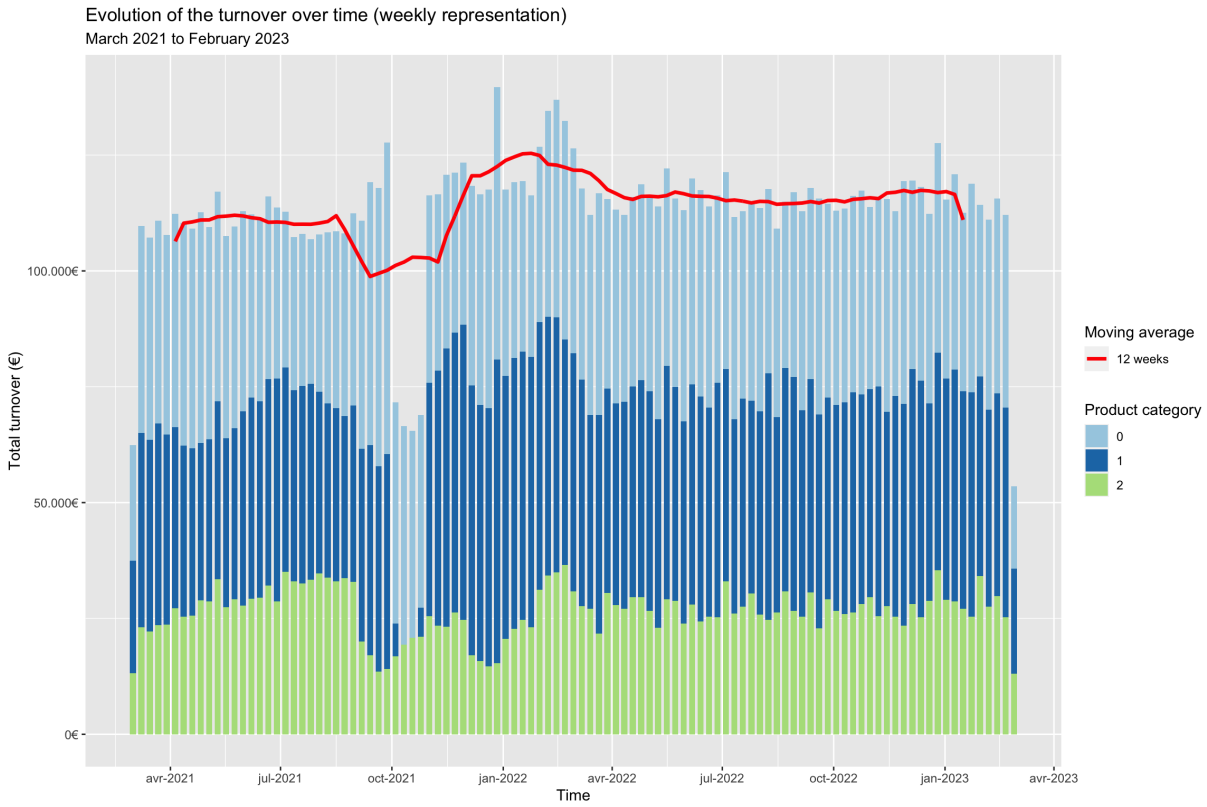
```
# We transform week 53 into 52 so as not to overestimate the cutting
globalCA$week[globalCA$week==53] <- 52

# Add a variable to retrieve a common date between the years, week
weekCA$yearWeekDay <- paste(weekCA$year,weekCA$week,'1', sep="")
globalCA$yearWeekDay <- paste(globalCA$year,globalCA$week,'1', sep="")
# Turn of the variable into date
weekCA$newDate <- as.Date(weekCA$yearWeekDay, "%Y%U%w")
globalCA$newDate <- as.Date(globalCA$yearWeekDay, "%Y%U%w")

# Turn of the obtained variables to gather the obtained days in a single line.
weekCA_2 <- weekCA %>%
  group_by(newDate, categ) %>%
  summarise(CA = sum(CA, na.rm=T))
```

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

```
globalCA_2 <- globalCA %>%
  group_by(newDate) %>%
  summarise(totalCA = sum(totalCA, na.rm=T))
```

```
# Creation of a dataframe for the moving average
rollmeanCA <- globalCA_2 %>% select(newDate,totalCA)
rollmeanCA <- rollmeanCA %>%
  mutate(CA_12da = rollmean(totalCA,k=12,fill=NA)) # 12 semaines
         #CA_05da = rollmean(totalCA,k=12,fill=NA))  # 5 semaines

chiffreAffaire <- left_join(weekCA_2,rollmeanCA, by="newDate") # join to obtain a single global data

# Pivot to retrieve data and put them in a single column for plot
ggplot(chiffreAffaire) +
  geom_bar(aes(x=newDate, y=CA, fill=categ), stat ='identity', width = 5)+
  scale_fill_brewer(palette="Paired")+
  geom_line(aes(x=newDate, y=CA_12da, colour = "12 weeks"),size=1.2) +
  #geom_line(aes(x=newDate, y=CA_05da, colour = "5 semaines"),size=1.2) +
  scale_color_manual(name = "Moving average", values = c("12 weeks" = "red"))+ #, "5 semaines" = "black")) +
  scale_y_continuous(labels = dollar_format(suffix = "€", prefix="", big.mark = '.', decimal.mark = ','))+
  scale_x_date(date_labels = ('%b-%Y'), date_breaks= "3 month")+
  labs(title="Evolution of the turnover over time (weekly representation)",
       subtitle = "March 2021 to February 2023",
       x="Time",
       y="Total turnover (€)",
       fill="Product category")
```

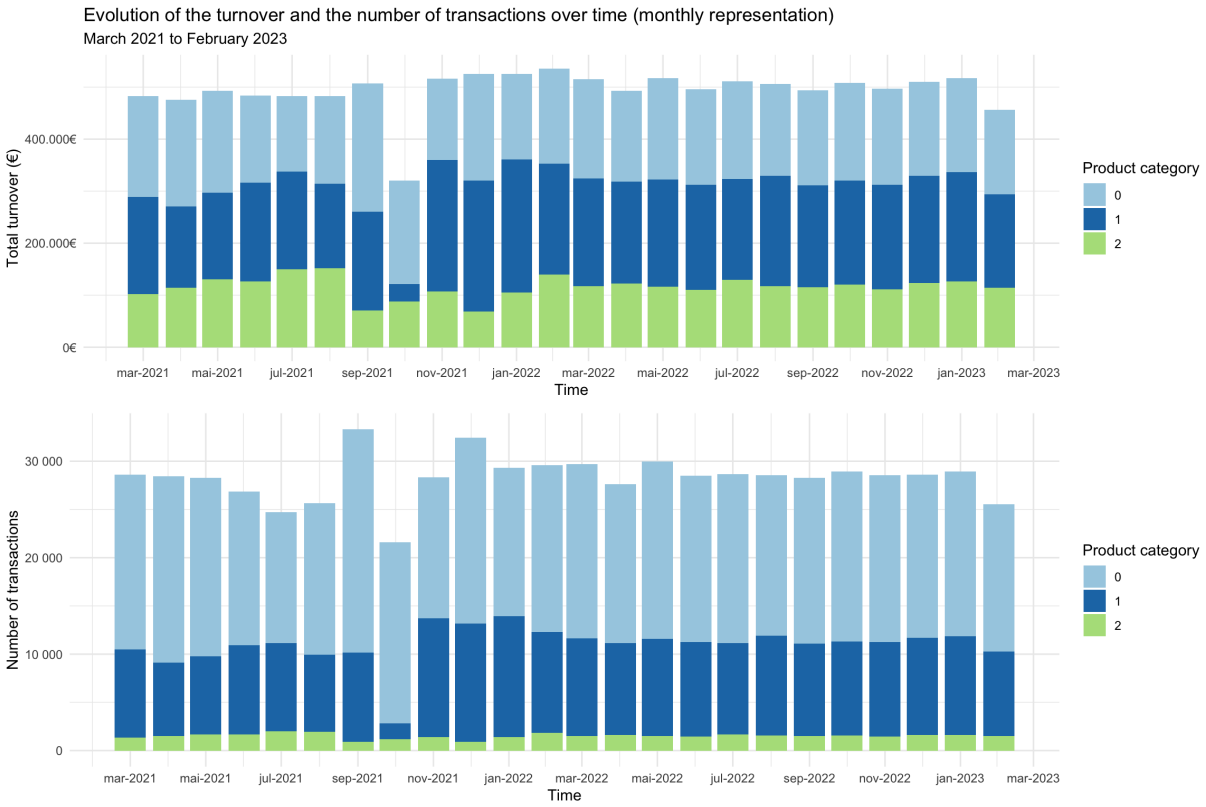
```
## Warning: Removed 33 row(s) containing missing values (geom_path).
```



We realize that the number of transactions for the month **of October 2021** are significantly lower than those of the previous month (September) and the next (November). Significant drop recorded for the month of October 2021, why?

Analysis of the month of October 2021

```
# Creation of a dataframe retrieving for each month and year the turnover and the number of transactions
monthsCA_categ <- df_final %>%
  mutate(month = sprintf("%02d",month)) %>%
  group_by(year,month,categ) %>%
  summarise(totalCAk = sum(price, na.rm=T),
            transaction = n())
# Add monthly date (Date format)
monthsCA_categ$yearMonth <- paste(monthsCA_categ$year,monthsCA_categ$month,'1',sep='-')
monthsCA_categ$date <- as.Date(monthsCA_categ$yearMonth, "%Y-%m-%d")

# Creation of a graph showing the evolution of the turnover and number of transacitons over time
g1 <- monthsCA_categ %>%
  ggplot()+
  geom_bar(aes(x=date, y=totalCAk,fill=categ), stat = 'identity')+
  scale_fill_brewer(palette="Paired")+
  scale_y_continuous(labels = dollar_format(suffix = "€", prefix="", big.mark = '.', decimal.mark = ','))+
  scale_x_date(date_labels = ('%b-%Y'), date_breaks= "2 month")+
  labs(title="Evolution of the turnover and the number of transactions over time (monthly representation)",
       subtitle = "March 2021 to February 2023",
       x="Time",
       y="Total turnover (€)",
       fill="Product category")+
  theme_minimal()

g2 <- monthsCA_categ %>%
  ggplot()+
  geom_bar(aes(x=date, y=transaction,fill=categ), stat = 'identity')+
  scale_fill_brewer(palette="Paired")+
  scale_y_continuous(labels = function(x) format(x, big.mark = ' '))+
  scale_x_date(date_labels = ('%b-%Y'), date_breaks= "2 month")+
  labs(x="Time",
       y="Number of transactions",
       fill="Product category")+
  theme_minimal()

grid.arrange(g1,g2,nrow=2)
```

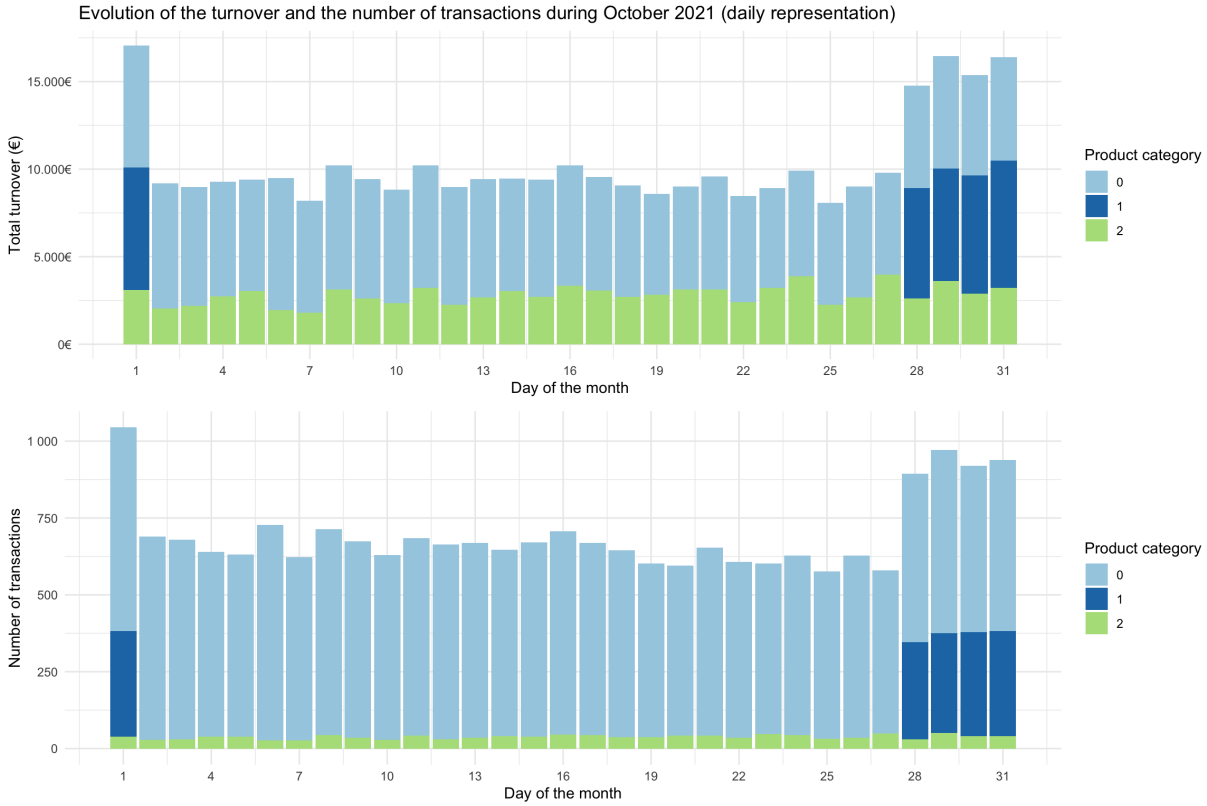


We have significantly fewer transactions for **Category 1** products for the month of October. We will inspect in detail.

```
# October 2021 Inspection
octobre2021 <- cbind(df_final)

# We recover only the year 2021 and the month 10 (October) of the final dataframe
octobre2021 <- octobre2021[(octobre2021$year == 2021) & (octobre2021$month == 10),]
infoOctobre2021 <- octobre2021 %>%
  group_by(day,categ) %>%
  summarise(CA = sum(price),
            transaction = n())
```

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

```
g1 <- infoOctobre2021 %>%
  ggplot()+
  geom_bar(aes(x=day,y=CA,fill=categ),stat='identity')+
  scale_fill_brewer(palette="Paired") +
  scale_y_continuous(labels = dollar_format(suffix = "€", prefix="", big.mark = '.', decimal.mark = ','))+
  scale_x_continuous(breaks= seq(1,31,by=3))+
  labs(title="Evolution of the turnover and the number of transactions during October 2021 (daily representatio
n)",
       x="Day of the month",
       y="Total turnover (€)",
       fill="Product category")+
  theme_minimal()

g2 <- infoOctobre2021 %>%
  ggplot()+
  geom_bar(aes(x=day, y=transaction,fill=categ), stat = 'identity')+
  scale_fill_brewer(palette="Paired")+
  scale_y_continuous(labels = function(x) format(x, big.mark = ' '))+
  scale_x_continuous(breaks= seq(1,31,by=3))+
  labs(x="Day of the month",
       y="Number of transactions",
       fill="Product category")+
  theme_minimal()

grid.arrange(g1,g2,nrow=2)
```



Between **2021-10-02** and **2021-10-27** no sales related to **category** 1 products were taken into account in the database. This is potentially linked to a stockout of category 1 products or to a computer bug. Several solutions are available to us.

We choose to delete all the data from October 2021 to avoid any problems in our analysis.

```
# Create a copy of the current df_final to keep all data
final_october <- cbind(df_final)

# Deletion of the month of October 2021 in df_final
df_final <- df_final[!(df_final$month == 10 & df_final$year == 2021),]
```

Zoom on the references

Analysis of the 'price' variable

```
# Function to get the "mode" of the variable
getmode <- function(val){
  uniqvalue <- unique(val)
  uniqvalue[which.max(tabulate(match(val,uniqvalue)))]
}

# Function to get a statistical summary :
stats_summary <- function (df){
  name <-c("Mean","Mode","Median","Corrected variance", "Corrected standard deviation", "Skewness", "Kurtosis")
  value <- c(mean(df,na.rm = T),getmode(df),median(df, na.rm = T),var(df, na.rm = T),sd(df, na.rm = T),skewness(d
f, na.rm = T),kurtosis(df, na.rm = T))
  df <- data.frame(name,value)
  df$value <- lapply(df$value, round, 2)
  print(df)
}
# Application of the function on the 'price' of the analyzed dataframe :
stats_summary(df_final$price)
```

```
##                          name  value
## 1                        Mean  17.54
## 2                        Mode  15.99
## 3                      Median  13.99
## 4           Corrected variance 336.37
## 5 Corrected standard deviation  18.34
## 6                    Skewness   5.38
## 7                    Kurtosis  46.03
```

The average price of products sold on the website is 17.54€.

Top and flop references

```
# Creation of a sub-dataframe to sort products by turnover
tab <- df_final %>%
  group_by(id_prod) %>%
  summarise(CA = sum(price),
            transaction = n())
tab <- left_join(tab,df_products, by="id_prod")

# Top 10 - Product
kable(head(tab[order(-tab$CA),],10)) %>%
  kable_styling(latex_options = 'striped')
```

id_prod	CA	transaction	price	categ
2_159	92265.68	632	145.99	2
2_135	67403.23	977	68.99	2
2_112	62840.10	930	67.57	2
2_102	58962.58	997	59.14	2
2_209	55362.09	791	69.99	2
1_395	53950.39	1861	28.99	1
1_369	53665.63	2237	23.99	1
2_110	51916.50	834	62.25	2
1_414	51615.78	2166	23.83	1
2_39	51147.18	882	57.99	2

```
# Bottom 10 - Product
kable(head(tab[order(tab$CA),],10))%>%
  kable_styling(latex_options = 'striped')
```

id_prod	CA	transaction	price	categ
0_1539	0.99	1	0.99	0
0_898	1.27	1	1.27	0
0_1284	1.38	1	1.38	0
0_1653	1.98	2	0.99	0
0_643	1.98	2	0.99	0
0_1601	1.99	1	1.99	0
0_541	1.99	1	1.99	0
0_807	1.99	1	1.99	0
0_1728	2.27	1	2.27	0
0_324	2.36	2	1.18	0

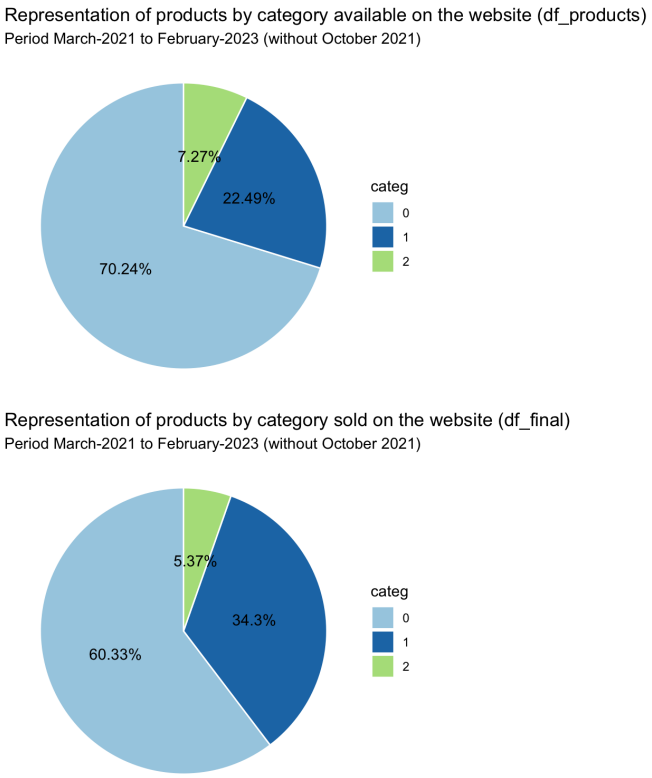
We can see that we have products with a turnover > 50.000€ with more than 2.000 products purchased. No product of category 0 is present in the top 10 products.

We realize that at least each product has been purchased.

Distribution by category

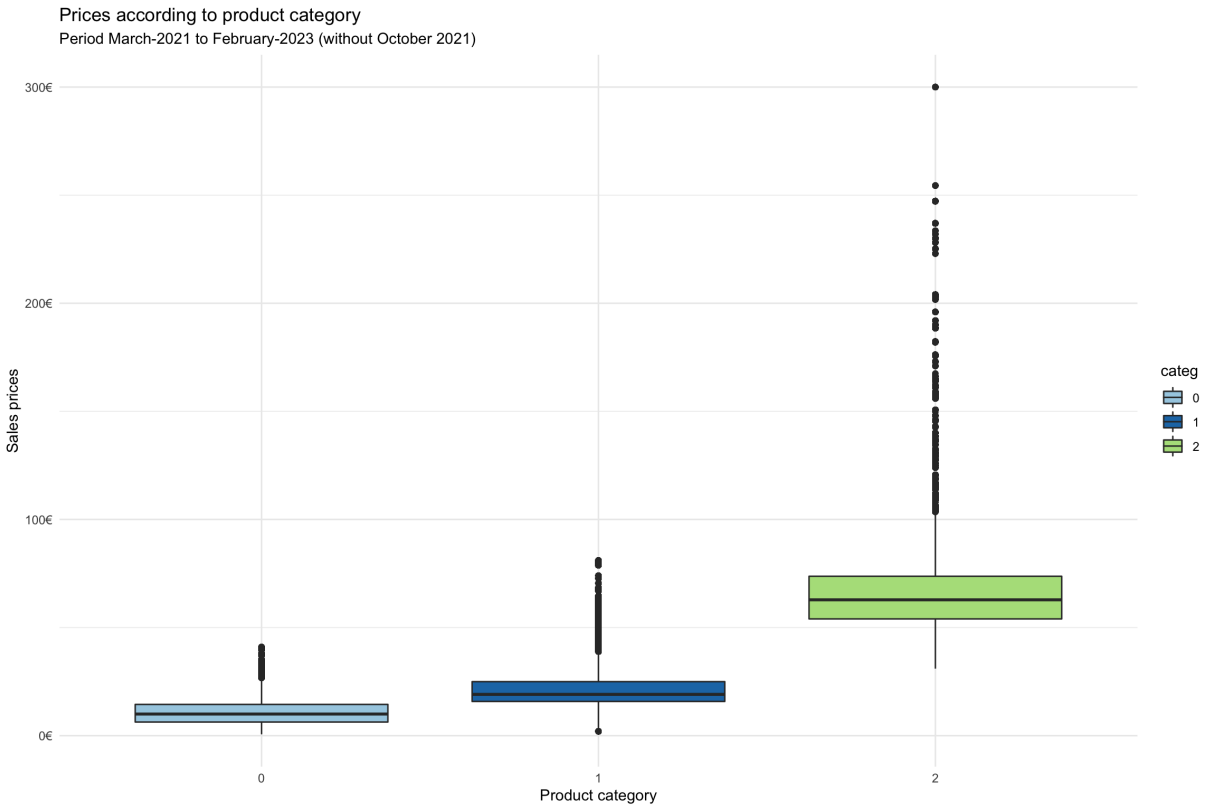
```
# Representation of products by category available on the website (df_products) :
tab <- df_products %>%
  group_by(categ)%>%
  summarise(transaction = n(),
            frequence = n()/nrow(df_products)) %>%
  mutate(frequence_c = cumsum(frequence),
         categ = as.factor(categ))

g1 <- tab %>%
ggplot(aes(x="",y=transaction,fill=categ))+
  geom_col(width=1, color="white")+
  coord_polar('y', start=0)+
  geom_text(aes(label = paste0(round(transaction/sum(transaction)*100,2),"%")),position = position_stack(vjust =
```

```
0.5))+
  labs(title = "Representation of products by category available on the website (df_products) ",
       subtitle = "Period March-2021 to February-2023 (without October 2021)")+
  theme_void()+
  scale_fill_brewer(palette="Paired")

# Representation of products by category sold on the website (df_final) :
tab <- df_final %>%
  group_by(categ)%>%
  summarise(transaction = n(),
            frequence = n()/nrow(df_final)) %>%
  mutate(frequence_c = cumsum(frequence))

g2 <- tab %>%
ggplot(aes(x="",y=transaction,fill=categ))+
  geom_col(width=1, color="white")+
  coord_polar('y', start=0)+
  geom_text(aes(label = paste0(round(transaction/sum(transaction)*100,2),"%")),position = position_stack(vjust =
0.5))+
  labs(title = "Representation of products by category sold on the website (df_final) ",
       subtitle = "Period March-2021 to February-2023 (without October 2021)")+
  theme_void()+
  scale_fill_brewer(palette="Paired")

grid.arrange(g1,g2,nrow=2)
```



Even if category 0 products are the most sold (> 60%). They are not among the top products in terms of turnover.

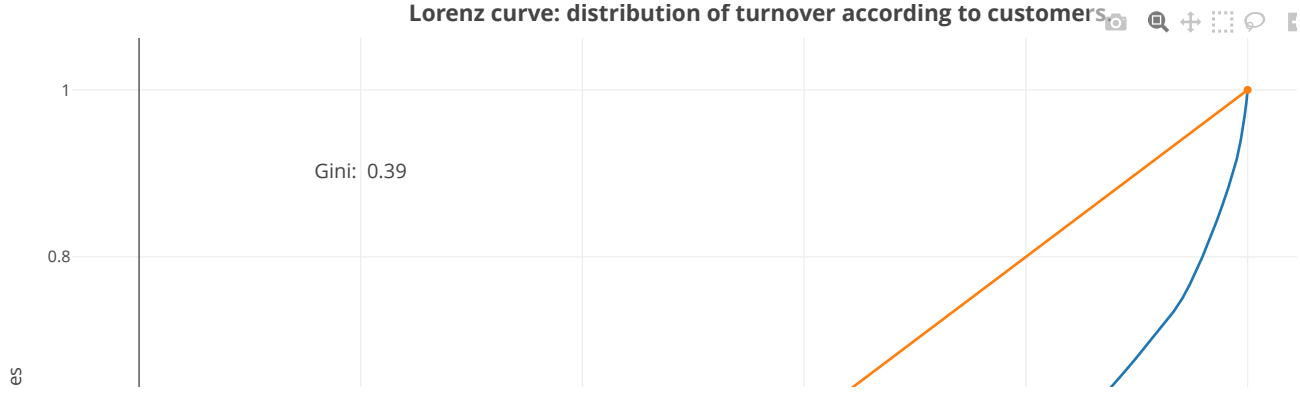
```
# Box-plot: Price/Product category
df_final %>%
  ggplot()+
  geom_boxplot(aes(x=categ,y=price,fill=categ,group=categ))+
  scale_fill_brewer(palette="Paired")+
  labs(title = "Prices according to product category",
       subtitle = "Period March-2021 to February-2023 (without October 2021)",
       x="Product category",
       y="Sales prices")+
  scale_y_continuous(labels = dollar_format(suffix = "€", prefix="", big.mark = '.', decimal.mark = ','))+
  theme_minimal()
```

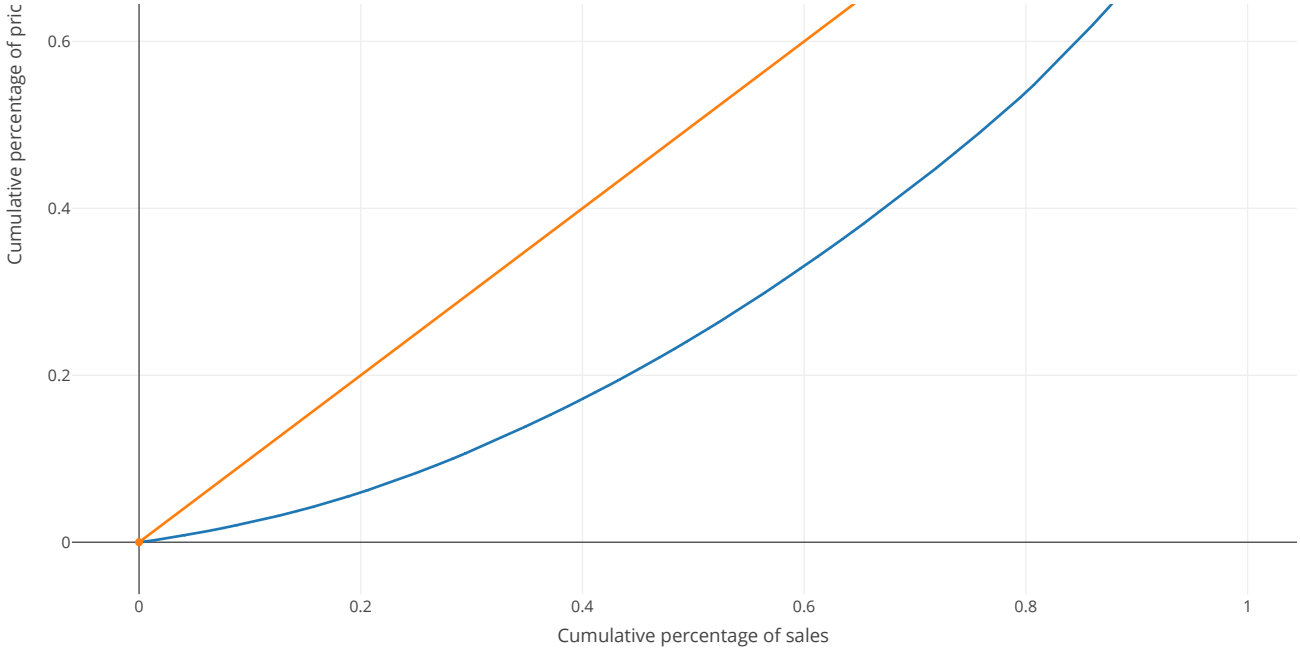


Thanks to the box-plot, we can see that the prices of the **category 2** products are higher than those of the categories 0 and 1.

Lorenz curve - Price

```
# Lorenz curve to show the distribution of the prices on the total turnover
tab <- Lc(df_final$price) # recovery of the points of the Lorenz curve

# Data construction
trace1 <- list(name = "Lorenz Curve",x = tab$p,y = tab$L)
trace2 <- list(name = "Line of equality", x = c(0, 1), y = c(0, 1))

# Plot design
layout <- list(
  title = "<b>Lorenz curve: distribution of turnover according to customers</b>.",
  xaxis = list(
    type = "linear",
    title = "Cumulative percentage of sales"
  ),
  yaxis = list(
    type = "linear",
    title = "Cumulative percentage of prices"
  ),
  autosize = TRUE
)
giniIndice <- paste("Gini: ",round(ineq(df_final$price,type="Gini"),2))

p <- plot_ly()
p <- add_trace(p, name=trace1$name, x=trace1$x, y=trace1$y, type = 'scatter', mode = 'lines')
p <- add_trace(p, name=trace2$name, x=trace2$x, y=trace2$y, type = 'scatter',mode = 'lines+markers')
p <- add_trace(p, x=0.2, y=0.9, type = 'scatter',mode = "text", text = giniIndice,textfont = list(size=15), showl
egend = FALSE)
p <- layout(p, title=layout$title, xaxis=layout$xaxis, yaxis=layout$yaxis, autosize=layout$autosize)
p
```

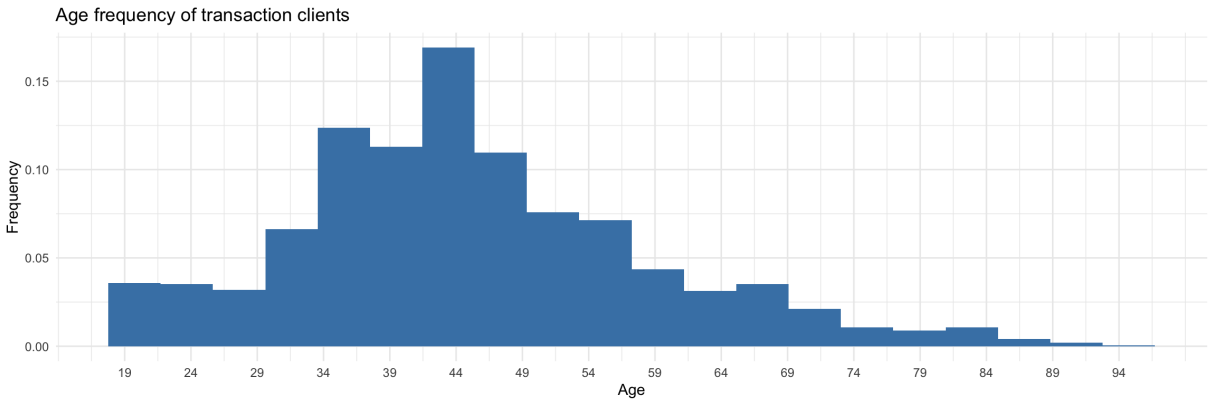


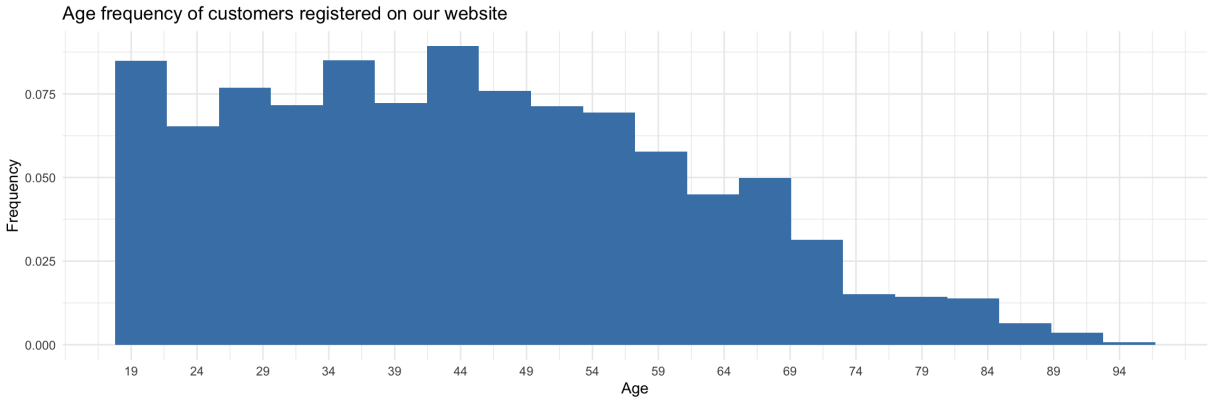


Client Information

Analysis of the variable 'birth

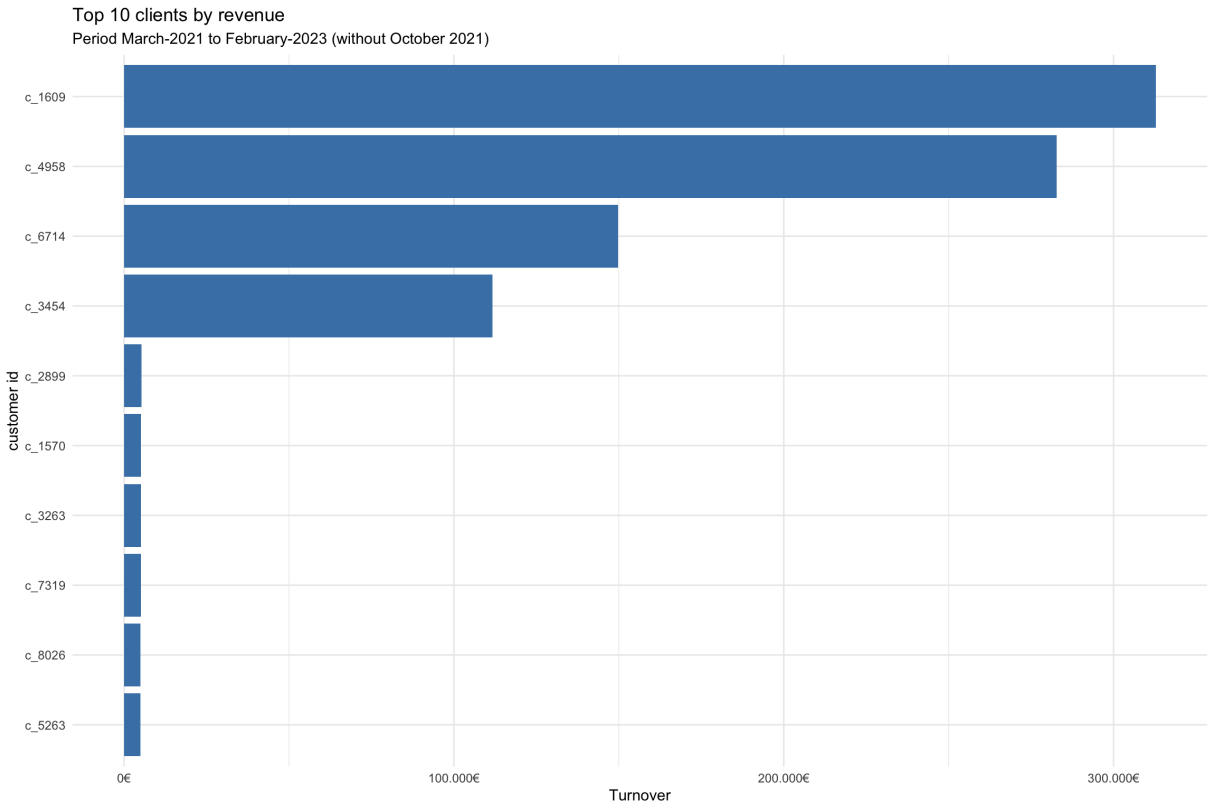
```
# Add an age variable to get the age of the client
df_final$age <- 2023 - df_final$birth

# Frequency of client age (df_final)
g1 <- df_final %>%
  ggplot(aes(x=age))+
  geom_histogram(aes(y=stat(count/sum(count))),bins=20,fill='steelblue')+
  labs(title="Age frequency of transaction clients",
       y = "Frequency",
       x = "Age")+
  scale_x_continuous(breaks= seq(min(df_final$age),max(df_final$age),by=5))+
  theme_minimal()

# Age frequency of customers (df_customers)
df_customers_copy <- cbind(df_customers)
df_customers_copy$age <- 2023 - df_customers_copy$birth

g2 <- df_customers_copy %>%
  ggplot(aes(x=age))+
  geom_histogram(aes(y=stat(count/sum(count))),bins=20,fill='steelblue')+
  labs(title="Age frequency of customers registered on our website",
       y = "Frequency",
       x = "Age")+
  scale_x_continuous(breaks= seq(min(df_customers_copy$age),max(df_customers_copy$age),by=5))+
  theme_minimal()

grid.arrange(g1,g2,nrow=2)
```





Comparison on turnover

```
# Table grouping the customer ids with the realized turnover
client_CA <- df_final %>%
  group_by(client_id) %>%
  summarise(CA = sum(price)) %>%
  arrange(CA)

# Top 10 clients
client_CA %>%
  arrange(desc(CA)) %>%
  slice(1:10) %>%
  ggplot()+
  geom_col(aes(x=CA,y= reorder(client_id, CA)),fill="steelblue")+
  labs(title="Top 10 clients by revenue",
       subtitle = "Period March-2021 to February-2023 (without October 2021)",
       y = "customer id",
       x = "Turnover")+
  scale_x_continuous(labels = dollar_format(suffix = "€", prefix="", big.mark = '.', decimal.mark = ','))+
  theme_minimal()
```




```
# Removal of potential companies
client_CA_2 <- client_CA %>%
  arrange(desc(CA))%>%
  slice(-c(1:4))
```

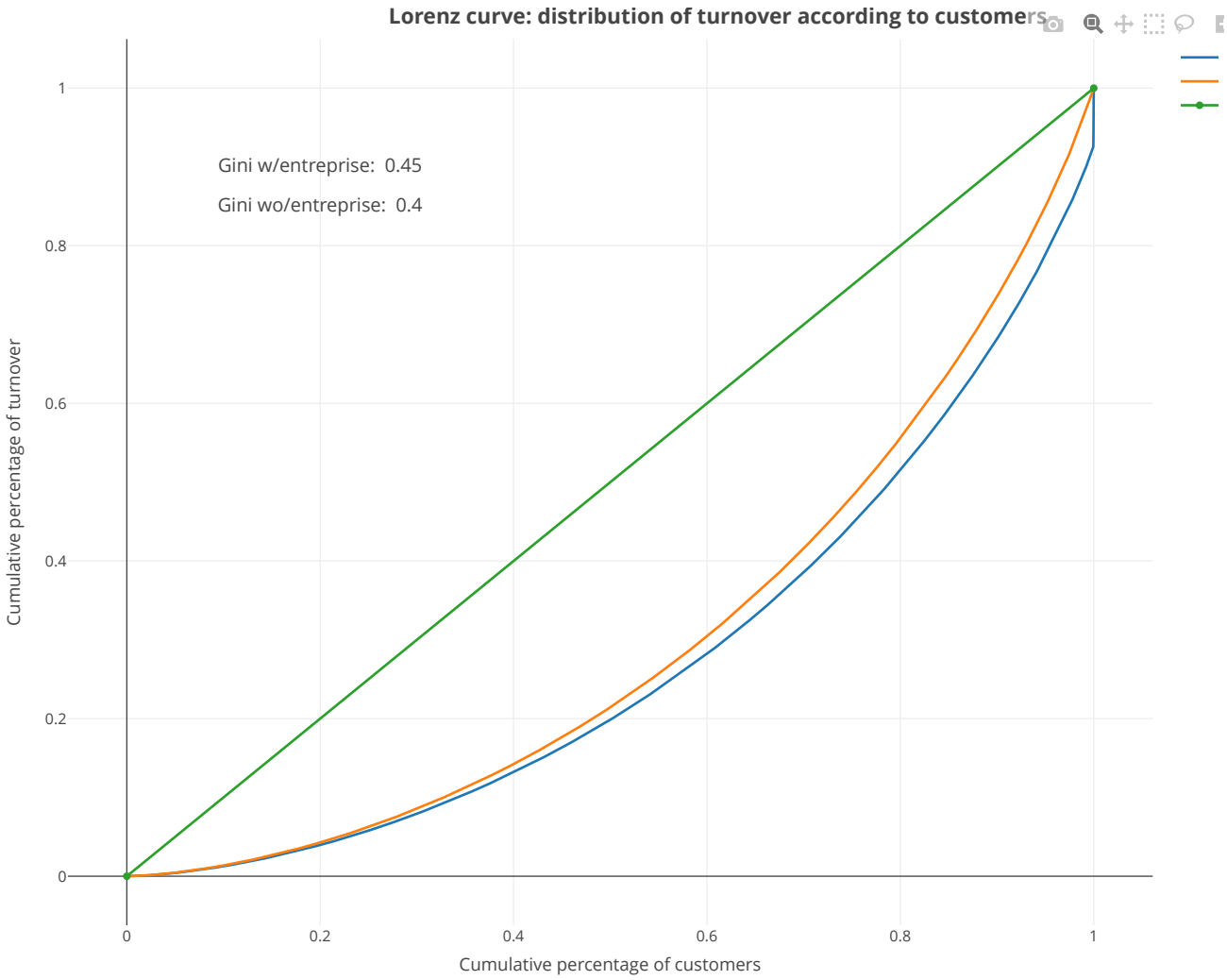
We realize that 4 customers have a **turnover** > 100.000€. These may be additional companies that have purchased equipment. Therefore, we have removed these 4 companies to compare the Lorenz curve - distribution of the turnover according to the customers and compare the 2 curves.

Moreover, for future analyses, it is interesting to remove them since the turnover can modify the analysis of our customers.

```
# Create a copy of the current dataframe
final_all_client <- cbind(df_final)

# We get the 4 client_id corresponding to the ID of the companies
client_outlier <- as.list(client_CA%>%
  arrange(desc(CA)) %>%
  slice(1:4) %>%
  select(client_id))

# We remove the client IDs of the companies from the df_final
df_final <- subset(df_final, !(client_id %in% client_outlier$client_id))
```

```
# Lorenz Curve to show the distribution of the turnover according to the customers
tab <- Lc(client_CA$CA) # ineq library to retrieve data from the Lorenz Curve
tab2 <- Lc(client_CA_2$CA)

# Data construction
trace1 <- list(name = "Lorenz Curve w/ entreprise",x = tab$p,y = tab$L)
trace2 <- list(name = "Line of equality", x = c(0, 1), y = c(0, 1))
trace3 <- list(name = "Lorenz Curve wo/ entreprise",x = tab2$p,y = tab2$L)

# Plot design
layout <- list(
  title = "<b>Lorenz curve: distribution of turnover according to customers</b>",
  xaxis = list(
    type = "linear",
    title = "Cumulative percentage of customers"
  ),
  yaxis = list(
    type = "linear",
    title = "Cumulative percentage of turnover"
  ),
  autosize = TRUE
)
giniIndice <- paste("Gini w/entreprise: ",round(ineq(client_CA$CA,type="Gini"),2))
giniIndice2 <- paste("Gini wo/entreprise: ",round(ineq(client_CA_2$CA,type="Gini"),2))

p <- plot_ly()
```

```
p <- add_trace(p, name=trace1$name, x=trace1$x, y=trace1$y, type = 'scatter', mode = 'lines')
p <- add_trace(p, name=trace3$name, x=trace3$x, y=trace3$y, type = 'scatter', mode = 'lines')
p <- add_trace(p, name=trace2$name, x=trace2$x, y=trace2$y, type = 'scatter',mode = 'lines+markers')
p <- add_trace(p, x=0.2, y=0.9, type = 'scatter',mode = "text", text = giniIndice,textfont = list(size=15), showl
egend = FALSE)
p <- add_trace(p, x=0.2, y=0.85, type = 'scatter',mode = "text", text = giniIndice2,textfont = list(size=15), sho
wlegend = FALSE)
p <- layout(p, title=layout$title, xaxis=layout$xaxis, yaxis=layout$yaxis, autosize=layout$autosize)
p
```



The Gini index decreased by removing the 4 customers that were potentially companies/resellers.

The 4 big customers represent almost 10% of our company's turnover. With companies: 20% of our customers represent 51% of our turnover
Without companies : 20% of our customers represent 44% of our turnover

Bonus : Loyalty analysis

```
# Products for each customer and transaction
tab <- df_final%>%
  group_by(client_id,session_id) %>%
  summarize(nProduit = n())
```

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

```
# transactions made for each customer
tab2 <- tab %>%
  group_by(client_id) %>%
  summarize(nTransaction = n())

# Table
Info <- c("More than one order", "One order")
NbClient <- c(nrow(tab2[tab2$nTransaction > 1,]),nrow(tab2[tab2$nTransaction == 1,]))
pourcentageClient <- NbClient/nrow(tab2)

fideliteClient <- data.frame(Info,NbClient,pourcentageClient)
kable(fideliteClient)%>%
  kable_styling(latex_options = 'striped')
```

Info	NbClient	pourcentageClient
More than one order	8559	0.9959274
One order	35	0.0040726

Only **35** customers have made only one order, which is less than 1% of our customers. We have more than 99% of our customers who have returned to make a purchase on our website.

Julie's mission:

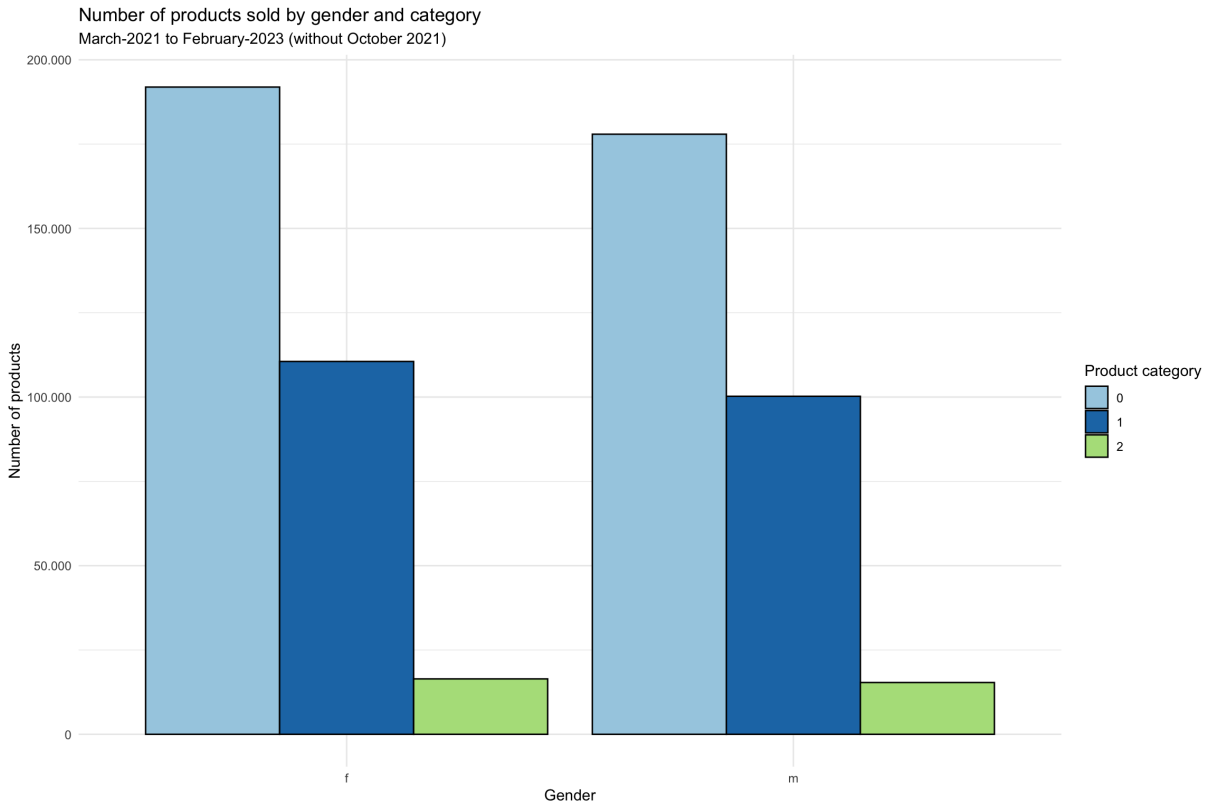
- Link between a customer's gender and the categories of books purchased;
- Link between the age of the customers and the total amount of the purchases,
- Frequency of purchase, average basket size and categories of books of books purchased

Relationship between customer gender and categories of books purchased:

```
# Distribution of purchases by category and gender
tab <- df_final %>%
  group_by(sex,categ) %>%
  summarize(CA = sum(price),
            nProduit = n())
```

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

```
ggplot(tab, aes(x=sex, y=nProduit, fill=categ))+
  geom_bar(stat='identity',color='black',position=position_dodge())+
  theme_minimal()+
  labs(title="Number of products sold by gender and category",
       subtitle = "March-2021 to February-2023 (without October 2021)",
       x="Gender",
       y="Number of products",
       fill="Product category")+
  scale_fill_brewer(palette="Paired")+
  scale_y_continuous(labels=function(x) format(x, big.mark = ".", decimal.mark = ",", scientific = FALSE))
```



Statistical test:
Variable:

- Gender -> qualitative
- Category -> qualitative

Test of association between two qualitative variables H0: There is no relationship between a customer's gender and the categories of books purchased (independent) H1 : There is a link between the gender of a customer and the categories of books purchased (dependent).

These are **independent** measures. Therefore, we will perform the chi-square test (association between two qualitative variables) on a contingency table.

```
# Creation of the contingency table
contingence_categ_sex <- table(df_final$categ, df_final$sex)
contingence_categ_sex
```

```
##
##          f      m
##   0 191919 177945
##   1 110550 100227
##   2  16429  15351
```

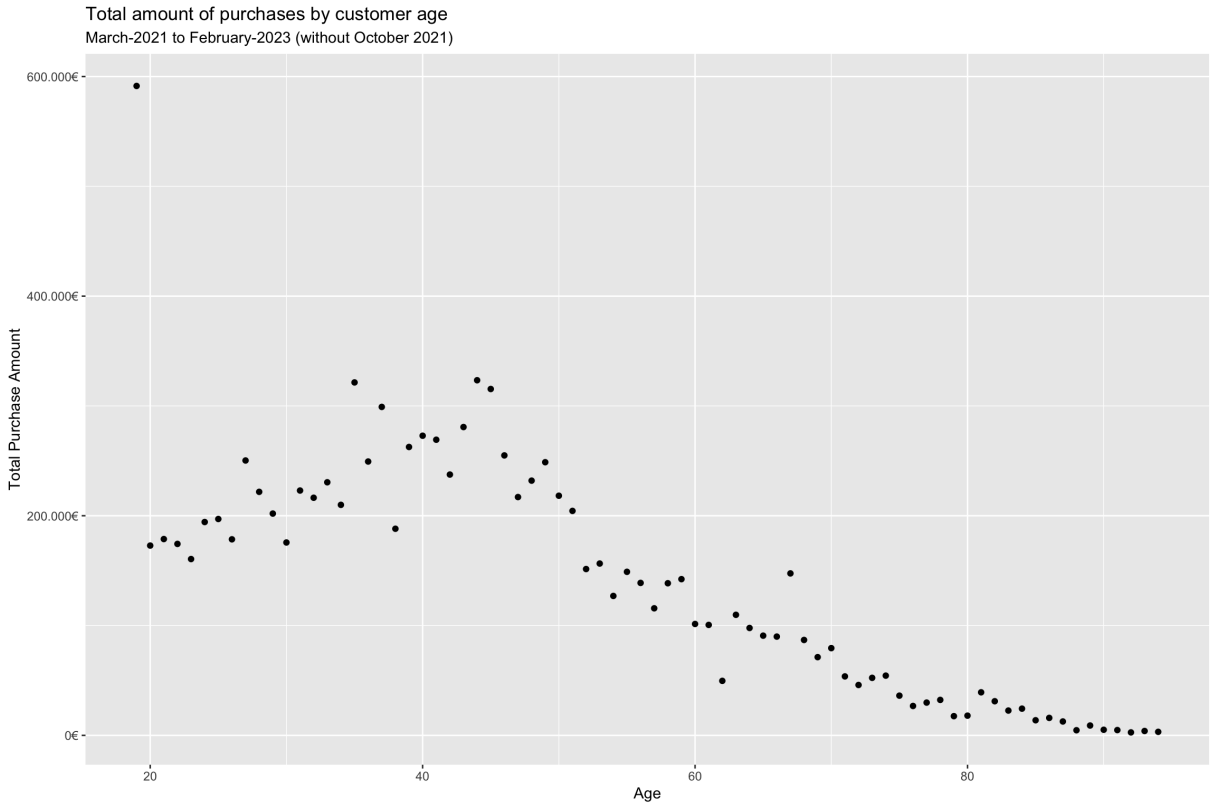
```
chisq.test(contingence_categ_sex)
```

```
##
##  Pearson's Chi-squared test
##
## data:  contingence_categ_sex
## X-squared = 18.75, df = 2, p-value = 8.483e-05
```

p-value <0.05 =>
we can consider that the variables are related. Thus, there is a link between the gender of a customer and the categories of books purchased.

Relationship between the age of the customers and the total amount of the purchases

```
# Age of customers and total amount of purchases
montantTotalAchat <- df_final %>%
  group_by(age) %>%
  summarize(totalCA = sum(price))

ggplot(montantTotalAchat, aes(x=age, y=totalCA))+
  geom_point()+
  labs(title="Total amount of purchases by customer age",
       subtitle = "March-2021 to February-2023 (without October 2021)",
       x="Age",
       y="Total Purchase Amount",
       color="Gender")+
  scale_y_continuous(labels = dollar_format(suffix = "€", prefix="", big.mark = '.', decimal.mark = ','))
```



Statistical test:
Variable:

- Age of customers: quantitative
- Total amount of purchases: quantitative

Test of association between two quantitative variables - verification of normality test H0: there is no correlation between the age of the customers and the total amount of purchases H1: there is a correlation between the age of the customers and the total amount of the purchases

We will use the Shapiro-Wilk test which is a test to know if a series of data follows a normal distribution or not.
Null hypothesis: the sample follows a normal distribution. Therefore if the p-value of the test is significant, the sample does not follow a normal distribution.

```
# Normality test - quantitative variable
shapiro.test(montantTotalAchat$totalCA)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  montantTotalAchat$totalCA
## W = 0.91196, p-value = 6.193e-05
```

```
shapiro.test(montantTotalAchat$age)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  montantTotalAchat$age
## W = 0.95492, p-value = 0.008753
```

p-value <0.05 => the variables do not follow a normal distribution. Use of the non-parametric test: **Spearman correlation**

```
# Spearman correlation
cor.test(montantTotalAchat$age,montantTotalAchat$totalCA,method='spearman')
```

```
##
##  Spearman's rank correlation rho
##
## data:  montantTotalAchat$age and montantTotalAchat$totalCA
## S = 137086, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.8740396
```

p-value <0.05 =>
Correlation between customer age and total purchase amount.
The correlation coefficient is -0.87.

Relationship between the age of the customers and the frequency of purchase

In our case, the purchase frequency corresponds to the number of transactions made per day by a specific age.

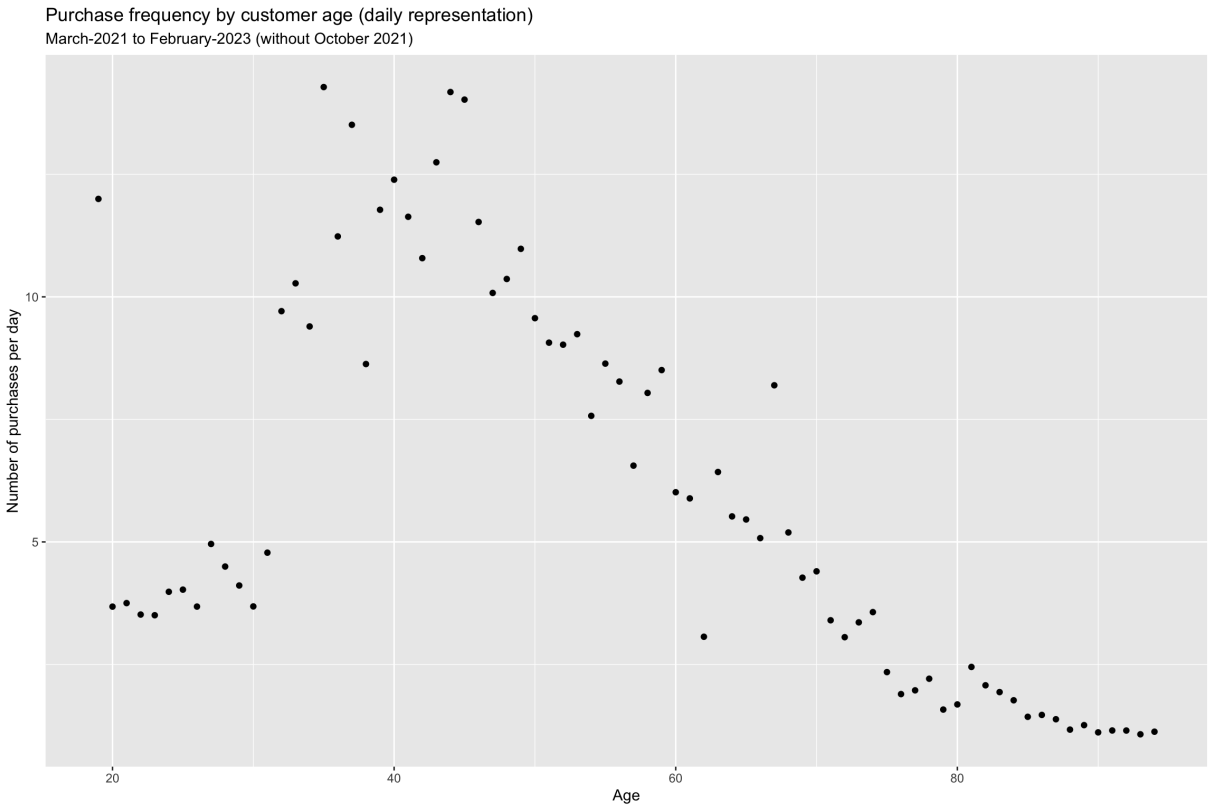
```
# Age of customers and daily purchase frequency
frequenceAchatProduit <- df_final %>%
  group_by(age,session_id, year, month, day) %>%
  summarize(nProduit = n())
```

```
## `summarise()` has grouped output by 'age', 'session_id', 'year', 'month'. You can override using the `.groups`
argument.
```

```
frequenceAchatTransaction <- frequenceAchatProduit %>%
  group_by(age, year, month, day) %>%
  summarize(nTransaction = n()) %>%
  group_by(age) %>%
  summarize(nTransactionMean = mean(nTransaction))
```

```
## `summarise()` has grouped output by 'age', 'year', 'month'. You can override using the `.groups` argument.
```

```
ggplot(frequenceAchatTransaction, aes(x=age,y=nTransactionMean))+geom_point()+
    labs(title="Purchase frequency by customer age (daily representation)",
         subtitle = "March-2021 to February-2023 (without October 2021)",
         x="Age",
         y="Number of purchases per day")
```



Statistical test:
Variable:

- Age of customers: quantitative
- Frequency of purchases: quantitative

Test of association between two quantitative variables - verification of normality test: H0: there is no correlation between the age of the customers and the frequency of purchases H1: there is a correlation between the age of the customers and the frequency of the purchases

We are going to use the Shapiro-Wilk test which is a test to know if a series of data follows a normal distribution or not.
Null hypothesis: the sample follows a normal distribution. Therefore if the p-value of the test is significant, the sample does not follow a normal distribution.

```
# Normality test - quantitative variable
shapiro.test(frequenceAchatTransaction$nTransactionMean)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  frequenceAchatTransaction$nTransactionMean
## W = 0.91527, p-value = 8.673e-05
```

```
shapiro.test(frequenceAchatTransaction$age)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  frequenceAchatTransaction$age
## W = 0.95492, p-value = 0.008753
```

p-value <0.05 => the variables do not follow a normal distribution. Use of the non-parametric test: **Spearman correlation**

```
# Spearman correlation:
cor.test(frequenceAchatTransaction$age,frequenceAchatTransaction$nTransactionMean,method='spearman',exact=FALSE)
```

```
##
##  Spearman's rank correlation rho
##
## data:  frequenceAchatTransaction$age and frequenceAchatTransaction$nTransactionMean
## S = 121086, p-value = 1.335e-10
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## -0.655311
```

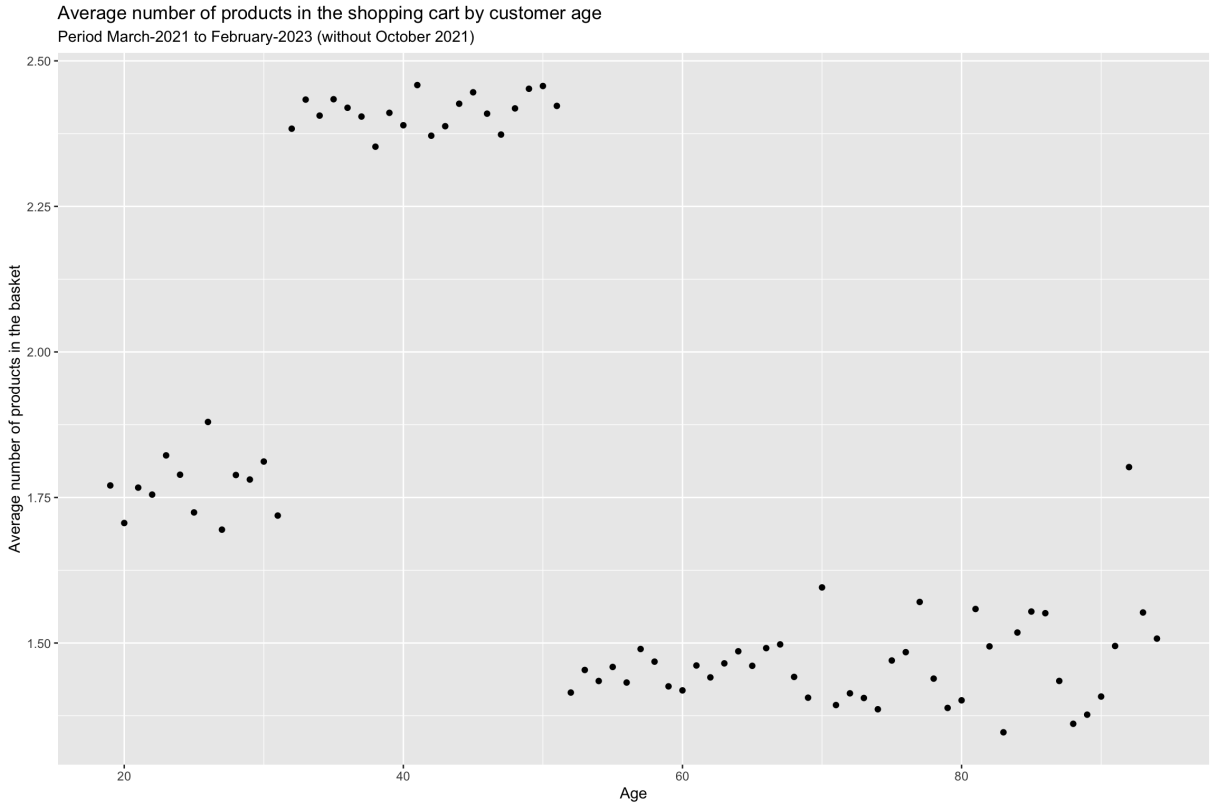
p-value <0.05 =>= correlation between customer age and daily shopping frequency correlation between customer age and daily shopping frequency. The correlation coefficient is -0.65.

Link between the age of the customers and the size of the average basket

```
# Customer age and average shopping cart size
panierMoyen <- df_final%>%
  group_by(age, session_id)%>%
  summarize(panier = n()) %>% # we get for each age and session the number of products
  group_by(age) %>%
  summarize(meanPanier = mean(panier)) # then group by age and average the products to get the average shopping c
art.
```

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

```
#Graph
ggplot(panierMoyen, aes(x=age,y=meanPanier))+geom_point()+
    labs(title="Average number of products in the shopping cart by customer age",
        subtitle = "Period March-2021 to February-2023 (without October 2021)",
         x="Age",
         y="Average number of products in the basket")
```



Statistical test:

Variable:

- Age of customers: quantitative
- Average shopping cart size: quantitative

Test of association between two quantitative variables - verification of normality test H0: there is no correlation between the age of the customers and the size of the average shopping cart H1: there is a correlation between the age of the customers and the size of the average shopping cart

We will use the Shapiro-Wilk test which is a test to know if a series of data follows a normal distribution or not.
Null hypothesis: the sample follows a normal distribution. Therefore if the p-value of the test is significant, the sample does not follow a normal distribution.

```
# Normality test
shapiro.test(panierMoyen$age)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  panierMoyen$age
## W = 0.95492, p-value = 0.008753
```

```
shapiro.test(panierMoyen$meanPanier)
```

```
##
##  Shapiro-Wilk normality test
##
```

```
## data:  panierMoyen$meanPanier
## W = 0.77204, p-value = 1.64e-09
```

p-value <0.05 => the variables do not follow a normal distribution. Use of the non-parametric test: **Spearman correlation**

```
# Spearman correlation
cor.test(panierMoyen$age,panierMoyen$meanPanier,method="spearman")
```

```
##
##  Spearman's rank correlation rho
##
## data:  panierMoyen$age and panierMoyen$meanPanier
## S = 118896, p-value = 1.883e-09
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## -0.6253725
```

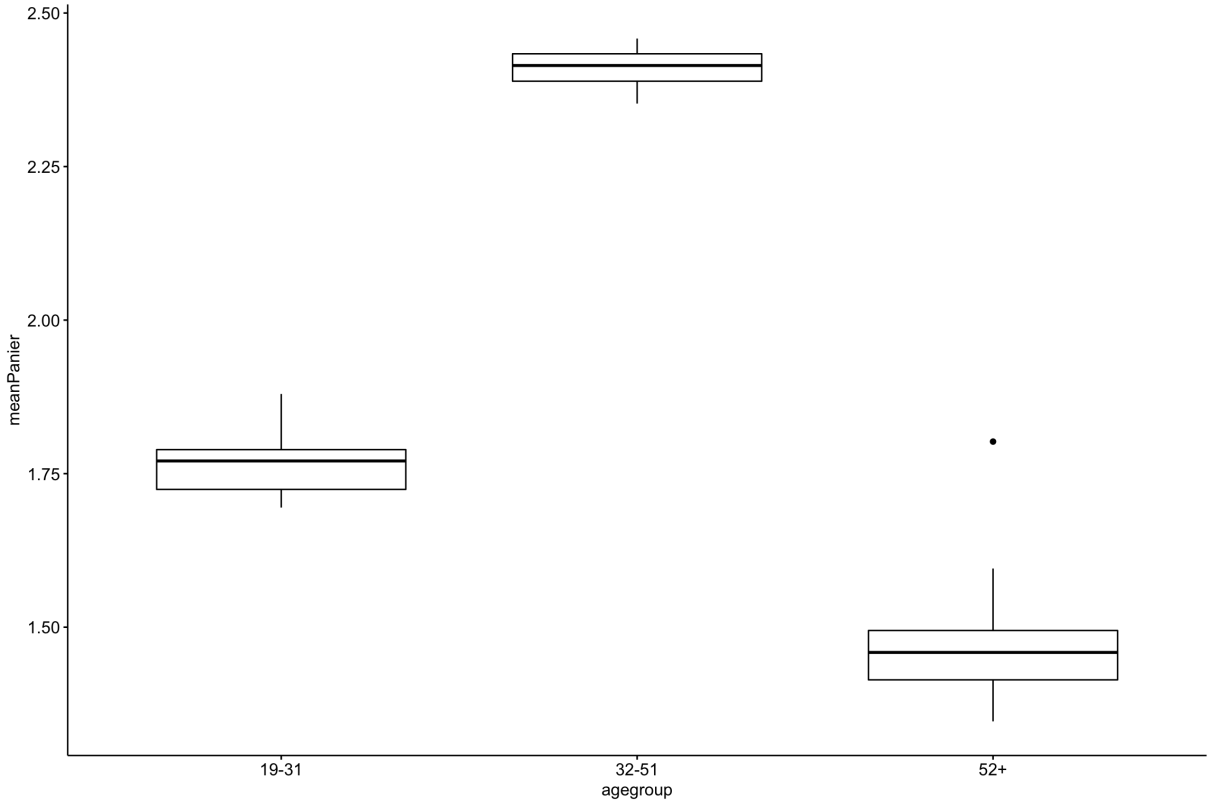
p-value <0.05 => correlation between customer age and average shopping cart. The correlation coefficient is -0.63.

We notice 3 distinct groups thanks to this graph. We will make a segmentation by age of these 3 groups, to check if there is a significant difference between these 3 groups on the average number of products in the shopping cart.

```
# Age segmentation according to the 3 groups obtained from the previous graph
panierMoyen <- panierMoyen %>%
  mutate(agegroup = case_when(age <= 31 ~ '19-31',
                              age >= 32 & age <= 51 ~'32-51',
                              age >= 52 ~'52+'))
# Descriptive Stats
panierMoyen %>%
  group_by(agegroup) %>%
  get_summary_stats(meanPanier, type ="common")
```

```
## # A tibble: 3 × 11
##   agegroup variable       n   min   max median   iqr  mean    sd    se    ci
##   <chr>    <chr>      <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 19-31    meanPanier    13  1.70  1.88   1.77 0.065  1.77 0.052 0.014 0.031
## 2 32-51    meanPanier    20  2.35  2.46   2.42 0.045  2.41 0.03  0.007 0.014
## 3 52+      meanPanier    43  1.35  1.80   1.46 0.08   1.47 0.079 0.012 0.024
```

```
# Visualization
ggboxplot(panierMoyen, x="agegroup", y="meanPanier")
```



Statistical test:

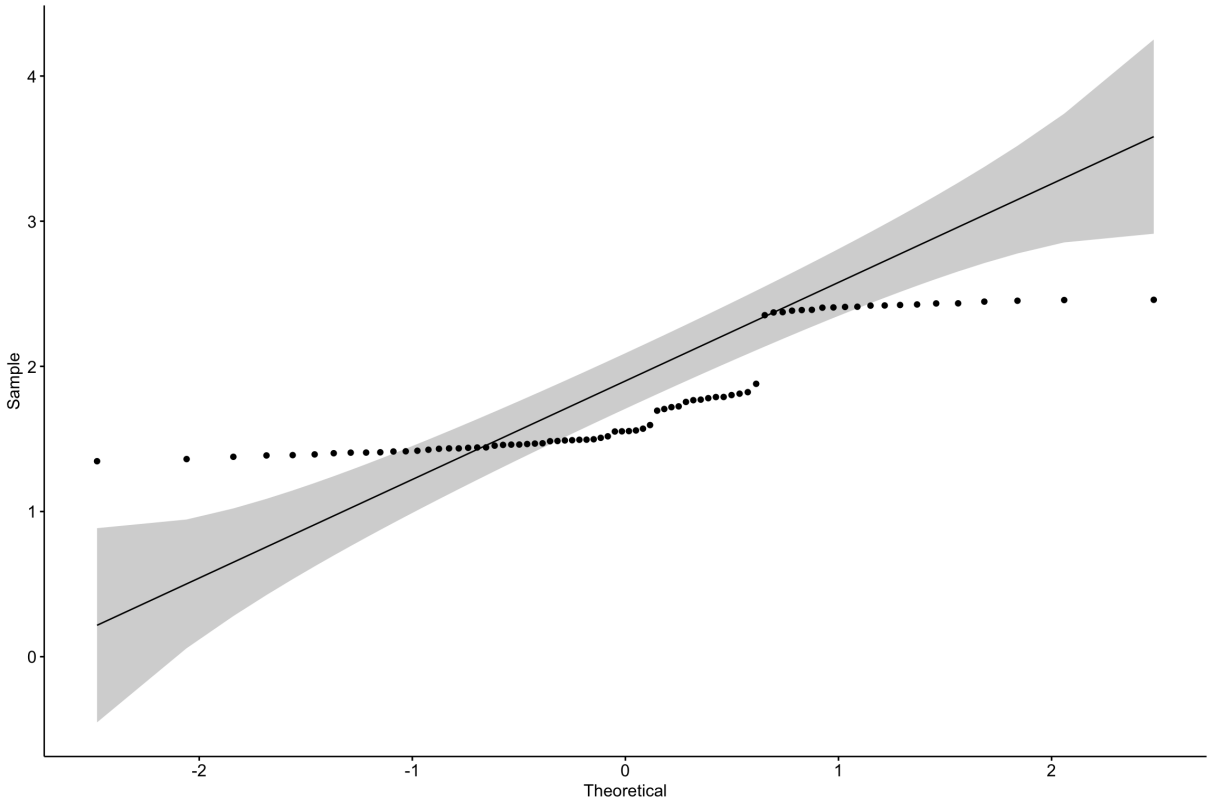
Variable:

- Age group: qualitative

- Average shopping cart: quantitative

One-way ANOVA: Analysis of a quantitative variable with a qualitative variable.
H0: There is no difference between the age group of the customers and the size of the average shopping cart H1: There is a difference between the age group of the customers and the size of the average shopping cart

```
# Normal Q-Q Plot
ggqqplot(panierMoyen$meanPanier)
```



We use the Normal Q-Q Plot to check the normality of the variable. The variable 'average shopping cart' does not follow a normal distribution, so we will use the Kruskal-Wallis (non-parametric test) instead of the ANOVA.

```
# Calculs
res.kruskal <- panierMoyen %>% kruskal_test(meanPanier ~ agegroup)
res.kruskal
```

```
## # A tibble: 1 × 6
##   .y.            n statistic    df        p method
## * <chr>      <int>     <dbl> <int>    <dbl> <chr>
## 1 meanPanier    76      58.6     2 1.93e-13 Kruskal-Wallis
```

```
# Effect size
panierMoyen %>% kruskal_effsize(meanPanier ~ agegroup)
```

```
## # A tibble: 1 × 5
##   .y.            n effsize method  magnitude
## * <chr>      <int>   <dbl> <chr>   <ord>
## 1 meanPanier    76   0.775 eta2[H] large
```

```
# Multiple pairwise comparison
pwc <- panierMoyen %>% wilcox_test(meanPanier ~ agegroup, p.adjust.method = "bonferroni")
pwc
```

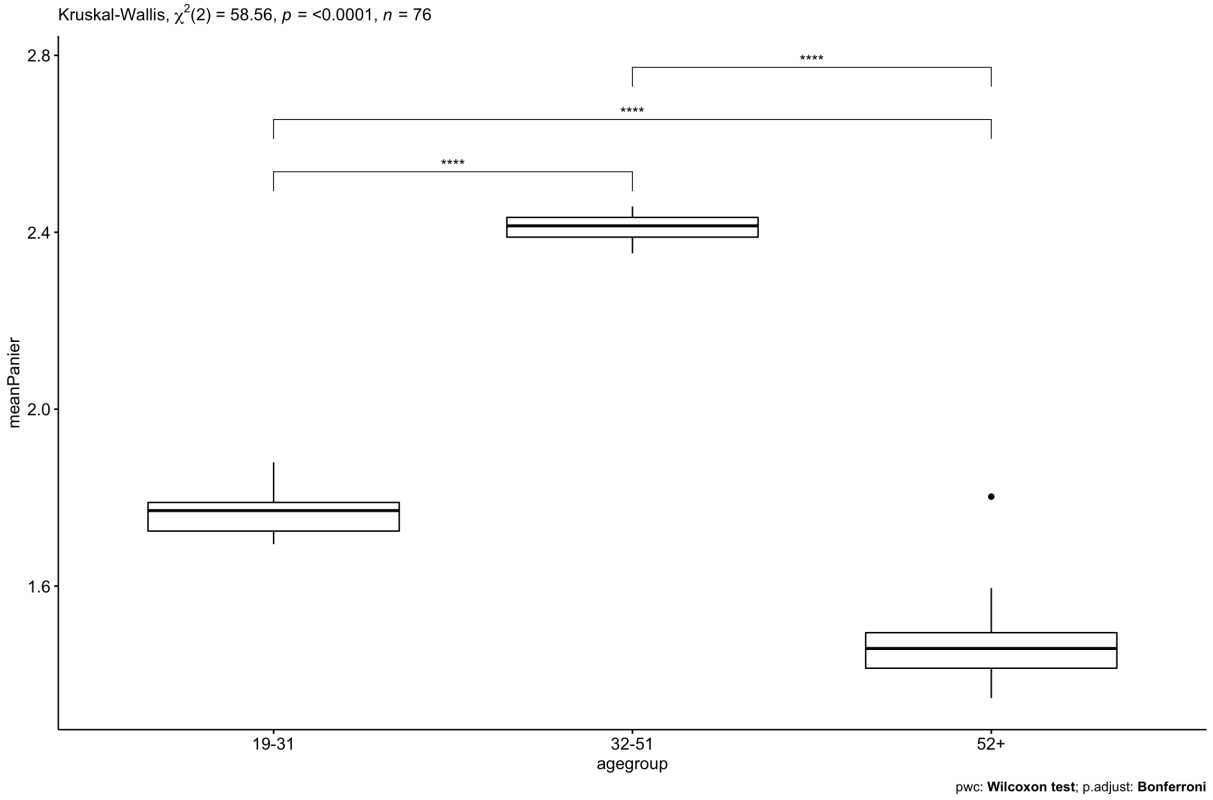
```
## # A tibble: 3 × 9
##   .y.        group1 group2    n1    n2 statistic        p    p.adj p.adj.signif
## * <chr>      <chr>  <chr>  <int> <int>     <dbl>    <dbl>    <dbl> <chr>
## 1 meanPanier 19-31  32-51     13    20         0 3.49e- 9 1.05e- 8 ****
## 2 meanPanier 19-31  52+       13    43       549 1.47e-10 4.41e-10 ****
## 3 meanPanier 32-51  52+       20    43       860 1.48e-16 4.44e-16 ****
```

p-value <0.05 =>
Difference between customer age group and average shopping cart.

Effect size = 0.775 (large)

To visualize the difference, we use the multiple pairwise comparison with the **Wilcoxon test**.

```
# Graphical representation
pwc <- pwc %>% add_xy_position(x = "agegroup")
ggboxplot(panierMoyen, x = "agegroup", y = "meanPanier") +
  stat_pvalue_manual(pwc, hide.ns = TRUE) +
  labs(
    subtitle = get_test_label(res.kruskal, detailed = TRUE),
    caption = get_pwc_label(pwc)
    )
```

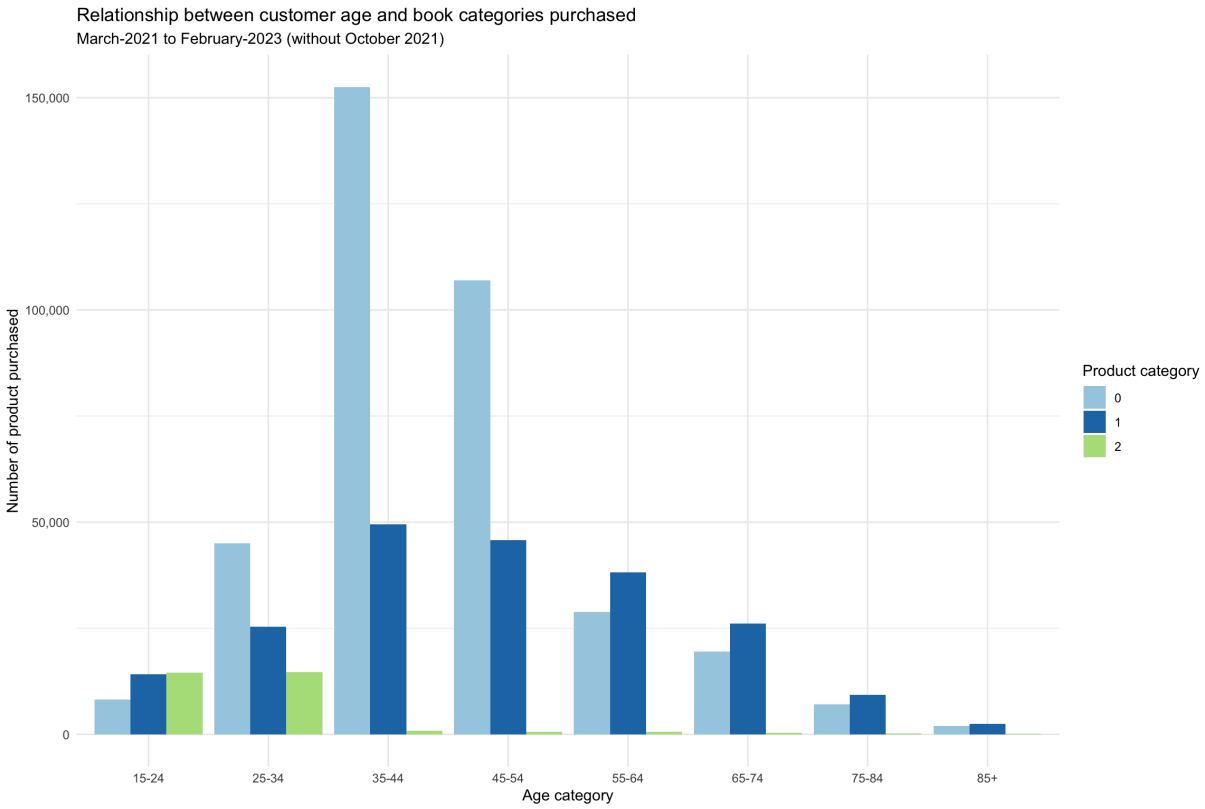


Relationship between customer age and categories of books purchased

```
# Creation of an age category data
labs <- c(paste(seq(15,84, by = 10),
                seq(15+10-1, 85-1, by=10),
                sep = "-"),
                paste(85,"+",sep=""))

df_final$groupeAge <- cut(df_final$age, breaks = c(seq(15,90,by = 10), Inf), labels = labs, right=FALSE)

ageCategorie <- df_final %>%
  group_by(groupeAge,categ)%>%
  summarize(nProduit = n())
```

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

```
# Graphic
ageCategorie %>%
  ggplot()+
  geom_col(aes(x=groupeAge,y=nProduit, fill=categ),position=position_dodge())+
  scale_fill_brewer(palette="Paired") +
  scale_y_continuous(labels = scales::comma)+
  labs(title="Relationship between customer age and book categories purchased",
       subtitle = "March-2021 to February-2023 (without October 2021)",
       x="Age category",
       y="Number of product purchased",
       fill="Product category")+
  theme_minimal()
```



Statistical test:
Variable:

- Age of customers: quantitative
- Category of books purchased: qualitative

One-way ANOVA: Analysis of a quantitative variable with a qualitative variable.

H0: there is no difference between the age of the customers and the category of books purchased H1 : there is a difference between the age of the customers and the category of books purchased

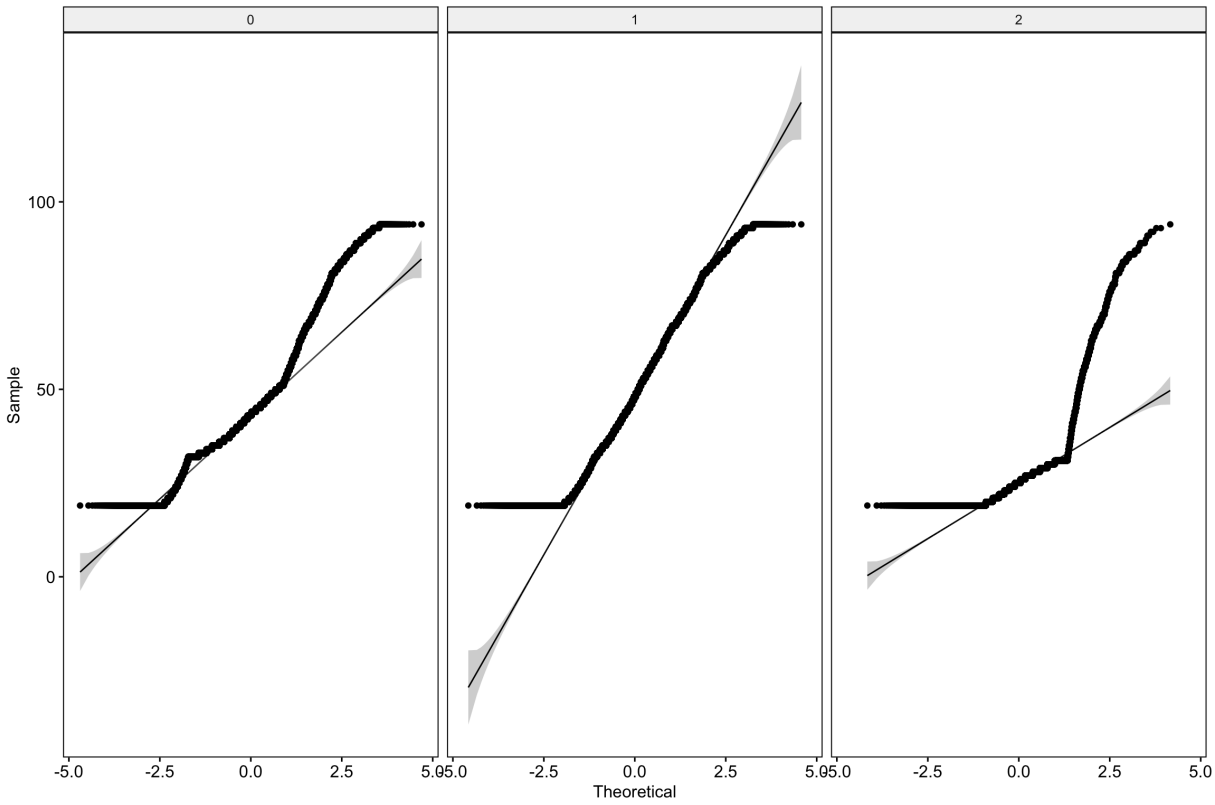
```
# We retrieve the information we are interested in, namely the age and the category associated in the final dataf
rame
analyseAnova <- df_final %>%
  select(age, categ)
levels(analyseAnova$categ)
```

```
## [1] "0" "1" "2"
```

```
# Descriptive statistics
analyseAnova %>%
  group_by(categ) %>%
  get_summary_stats(age, type = 'common')
```

```
## # A tibble: 3 × 11
##   categ variable      n   min   max median   iqr  mean    sd    se    ci
##   <fct> <chr>     <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 0     age      369864    19    94     43    12  44.8 11.5  0.019 0.037
## 2 1     age      210777    19    94     48    23  48.8 15.8  0.034 0.067
## 3 2     age       31780    19    94     25     8  27.0  9.90 0.056 0.109
```

```
# Visualization
ggqqplot(analyseAnova, 'age', facet.by = 'categ')
```



We use the Normal Q-Q Plot to check the normality of the variable. The age variable does not follow a normal distribution, so we will use the Kruskal-Wallis (non-parametric test) instead of the ANOVA.

```
# Computation
res.kruskal <- analyseAnova %>% kruskal_test(age ~ categ)
res.kruskal
```

```
## # A tibble: 1 × 6
##   .y.        n statistic    df     p method
## * <chr>  <int>     <dbl> <int> <dbl> <chr>
## 1 age   612421    69906.     2     0 Kruskal-Wallis
```

```
analyseAnova %>% kruskal_effsize(age ~ categ)
```

```
## # A tibble: 1 × 5
##   .y.        n effsize method  magnitude
## * <chr>  <int>   <dbl> <chr>   <ord>
## 1 age   612421   0.114 eta2[H] moderate
```

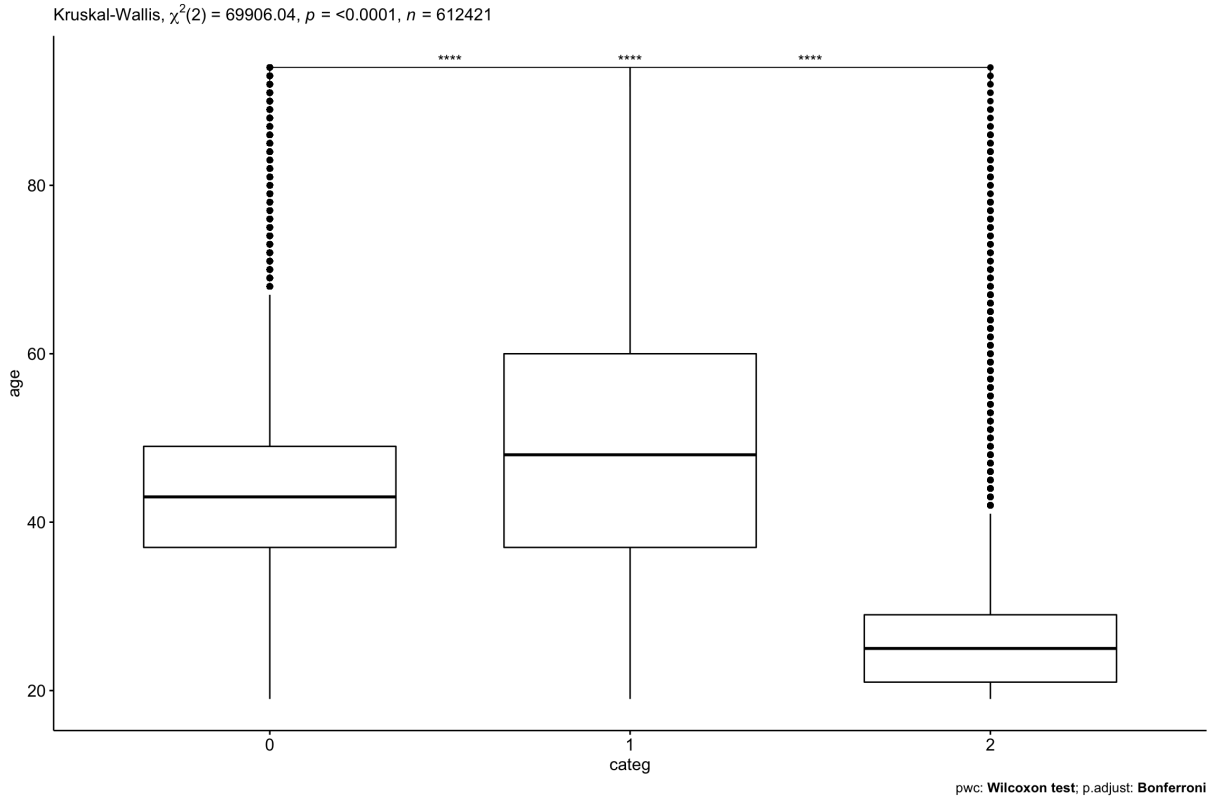
```
pwc <- analyseAnova %>%
  wilcox_test(age ~ categ, p.adjust.method = "bonferroni")
```

p-value <0.05 =>
Difference between customer age and book categories purchased.

Effect size = 0.114 (moderate)

To visualize the difference, we use the multiple pairwise comparison with the **Wilcoxon test**.

```
# Report
pwc <- pwc %>% add_xy_position(x = "categ")
ggboxplot(analyseAnova, x = "categ", y = "age") +
  stat_pvalue_manual(pwc, hide.ns = TRUE) +
  labs(
    subtitle = get_test_label(res.kruskal, detailed = TRUE),
    caption = get_pwc_label(pwc)
    )
```



End of Markdown R. Thank you for reading.