Marketing analysis

I- General introduction

Origin The present dataframe has been created for marketing analysis purposes. It assembles various personal information about 2239 customers, such as their education level, income, age, marital status, number of children at home... It also shows their consuming habits (amount spent on wine, on sweets...) and the number of purchases made on discounted products.

There is very few context concerning this dataframe, since the source is unknown. It is not clear when these informations were registered, but probably by 2014 since the date of customers' enrollment within the company doesn't go further than 2014.

Aims *To predict the customer's behavior (Number of purchases made with a discount) depending on the most significant personal attributes* To categorize participants in a few typical profiles (probably with PCA)

Attributes

- *People*

ID: Customer's unique identifier Year_Birth: Customer's birth year Education: Customer's education level Marital_Status: Customer's marital status Income: Customer's yearly household income Kidhome: Number of children in customer's household Teenhome: Number of teenagers in customer's household Dt_Customer: Date of customer's enrollment with the company Recency: Number of days since customer's last purchase Complain: 1 if customer complained in the last 2 years, 0 otherwise

- *Products*

MntWines: Amount spent on wine in last 2 years MntFruits: Amount spent on fruits in last 2 years MntMeatProducts: Amount spent on meat in last 2 years MntFishProducts: Amount spent on fish in last 2 years MntSweetProducts: Amount spent on sweets in last 2 years MntGoldProds: Amount spent on gold in last 2 years

- *Promotions*

NumDealsPurchases: Number of purchases made with a discount AcceptedCmp1: 1 if customer accepted the offer in the 1st campaign, 0 otherwise AcceptedCmp2: 1 if customer accepted the offer in the 2nd campaign, 0 otherwise AcceptedCmp3: 1 if customer accepted the offer in the 3rd campaign, 0 otherwise AcceptedCmp4: 1 if customer accepted the offer in the 4th campaign, 0 otherwise AcceptedCmp5: 1 if customer accepted the offer in the 5th campaign, 0 otherwise Response: 1 if customer accepted the offer in the last campaign, 0 otherwise NumStorePurchases: Number of purchases made directly in stores

Before loading the dataset, we want to make sure we have all the necessary packages installed and loaded, and that the code can be run by anybody.

```
if(!require(pacman)) {
  install.packages("pacman")
  library(pacman)
}
```

```
## Loading required package: pacman
```

```
pacman::p_load(tidyverse, gtsummary, ggpubr, moments, here, sjPlot, parameters, effectsize)

path =  here("JULIETTE")
setwd(path)
data <- read.table("marketing_campaign.csv", header=T, sep="\t")
```

II - Data overview and clearing

```
summary(data)
```

```
##        ID          Year_Birth    Education         Marital_Status
##  Min.   :    0   Min.   :1893   Length:2240        Length:2240
##  1st Qu.: 2828   1st Qu.:1959   Class :character   Class :character
##  Median : 5458   Median :1970   Mode  :character   Mode  :character
##  Mean   : 5592   Mean   :1969
##  3rd Qu.: 8428   3rd Qu.:1977
##  Max.   :11191   Max.   :1996
##
##      Income          Kidhome          Teenhome      Dt_Customer
##  Min.   :  1730   Min.   :0.0000   Min.   :0.0000   Length:2240
##  1st Qu.: 35303   1st Qu.:0.0000   1st Qu.:0.0000   Class :character
##  Median : 51382   Median :0.0000   Median :0.0000   Mode  :character
##  Mean   : 52247   Mean   :0.4442   Mean   :0.5062
##  3rd Qu.: 68522   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :666666   Max.   :2.0000   Max.   :2.0000
##  NA's   :24
##     Recency         MntWines         MntFruits     MntMeatProducts
##  Min.   : 0.00   Min.   :   0.00   Min.   :  0.0   Min.   :   0.0
##  1st Qu.:24.00   1st Qu.:  23.75   1st Qu.:  1.0   1st Qu.:  16.0
##  Median :49.00   Median : 173.50   Median :  8.0   Median :  67.0
##  Mean   :49.11   Mean   : 303.94   Mean   : 26.3   Mean   : 166.9
##  3rd Qu.:74.00   3rd Qu.: 504.25   3rd Qu.: 33.0   3rd Qu.: 232.0
##  Max.   :99.00   Max.   :1493.00   Max.   :199.0   Max.   :1725.0
##
##  MntFishProducts  MntSweetProducts  MntGoldProds    NumDealsPurchases
##  Min.   :  0.00   Min.   :  0.00   Min.   :  0.00   Min.   : 0.000
##  1st Qu.:  3.00   1st Qu.:  1.00   1st Qu.:  9.00   1st Qu.: 1.000
##  Median : 12.00   Median :  8.00   Median : 24.00   Median : 2.000
##  Mean   : 37.53   Mean   : 27.06   Mean   : 44.02   Mean   : 2.325
##  3rd Qu.: 50.00   3rd Qu.: 33.00   3rd Qu.: 56.00   3rd Qu.: 3.000
##  Max.   :259.00   Max.   :263.00   Max.   :362.00   Max.   :15.000
##
##  NumWebPurchases  NumCatalogPurchases NumStorePurchases NumWebVisitsMonth
##  Min.   : 0.000   Min.   : 0.000      Min.   : 0.00     Min.   : 0.000
##  1st Qu.: 2.000   1st Qu.: 0.000      1st Qu.: 3.00     1st Qu.: 3.000
##  Median : 4.000   Median : 2.000      Median : 5.00     Median : 6.000
##  Mean   : 4.085   Mean   : 2.662      Mean   : 5.79     Mean   : 5.317
##  3rd Qu.: 6.000   3rd Qu.: 4.000      3rd Qu.: 8.00     3rd Qu.: 7.000
##  Max.   :27.000   Max.   :28.000      Max.   :13.00     Max.   :20.000
##
##   AcceptedCmp3      AcceptedCmp4      AcceptedCmp5      AcceptedCmp1
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000   Median :0.00000   Median :0.00000   Median :0.00000
##  Mean   :0.07277   Mean   :0.07455   Mean   :0.07277   Mean   :0.06429
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000
##
##   AcceptedCmp2        Complain        Z_CostContact   Z_Revenue
##  Min.   :0.00000   Min.   :0.000000   Min.   :3     Min.   :11
```

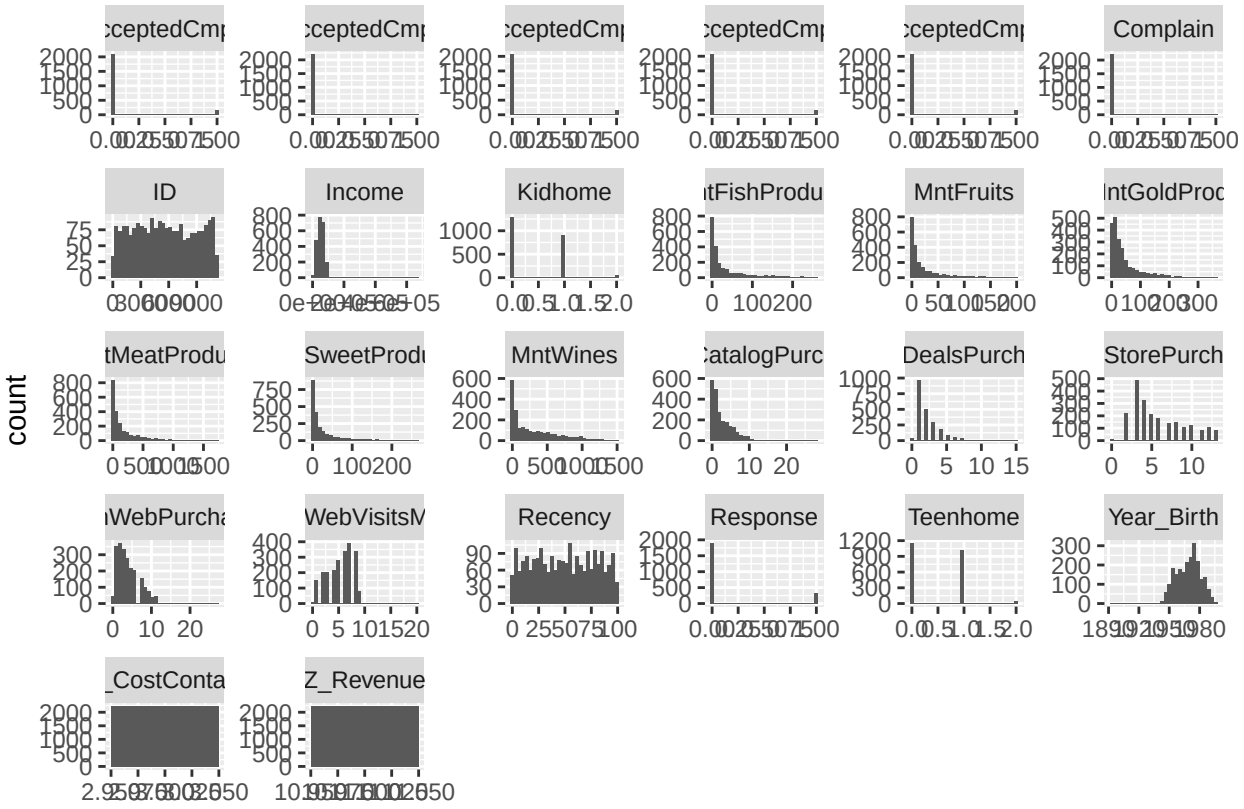
```
##  1st Qu.:0.00000   1st Qu.:0.000000   1st Qu.:3     1st Qu.:11
##  Median :0.00000   Median :0.000000   Median :3     Median :11
##  Mean   :0.01339   Mean   :0.009375   Mean   :3     Mean   :11
##  3rd Qu.:0.00000   3rd Qu.:0.000000   3rd Qu.:3     3rd Qu.:11
##  Max.   :1.00000   Max.   :1.000000   Max.   :3     Max.   :11
##
##     Response
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.1491
##  3rd Qu.:0.0000
##  Max.   :1.0000
##
```

```
str(data)
```

```
## 'data.frame':    2240 obs. of  29 variables:
##  $ ID                 : int  5524 2174 4141 6182 5324 7446 965 6177 4855 5899 ...
##  $ Year_Birth         : int  1957 1954 1965 1984 1981 1967 1971 1985 1974 1950 ...
##  $ Education          : chr  "Graduation" "Graduation" "Graduation" "Graduation" ...
##  $ Marital_Status     : chr  "Single" "Single" "Together" "Together" ...
##  $ Income             : int  58138 46344 71613 26646 58293 62513 55635 33454 30351 5648 ...
##  $ Kidhome            : int  0 1 0 1 1 0 0 1 1 1 ...
##  $ Teenhome           : int  0 1 0 0 0 1 1 0 0 1 ...
##  $ Dt_Customer        : chr  "04-09-2012" "08-03-2014" "21-08-2013" "10-02-2014" ...
##  $ Recency            : int  58 38 26 26 94 16 34 32 19 68 ...
##  $ MntWines           : int  635 11 426 11 173 520 235 76 14 28 ...
##  $ MntFruits          : int  88 1 49 4 43 42 65 10 0 0 ...
##  $ MntMeatProducts    : int  546 6 127 20 118 98 164 56 24 6 ...
##  $ MntFishProducts    : int  172 2 111 10 46 0 50 3 3 1 ...
##  $ MntSweetProducts   : int  88 1 21 3 27 42 49 1 3 1 ...
##  $ MntGoldProds       : int  88 6 42 5 15 14 27 23 2 13 ...
##  $ NumDealsPurchases  : int  3 2 1 2 5 2 4 2 1 1 ...
##  $ NumWebPurchases    : int  8 1 8 2 5 6 7 4 3 1 ...
##  $ NumCatalogPurchases: int  10 1 2 0 3 4 3 0 0 0 ...
##  $ NumStorePurchases  : int  4 2 10 4 6 10 7 4 2 0 ...
##  $ NumWebVisitsMonth  : int  7 5 4 6 5 6 6 8 9 20 ...
##  $ AcceptedCmp3       : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ AcceptedCmp4       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ AcceptedCmp5       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ AcceptedCmp1       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ AcceptedCmp2       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Complain           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Z_CostContact      : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ Z_Revenue          : int  11 11 11 11 11 11 11 11 11 11 ...
##  $ Response           : int  1 0 0 0 0 0 0 0 1 0 ...
```

```
data %>%
  keep(is.numeric) %>%
  gather() %>%
  ggplot(aes(value)) +
  facet_wrap(~key, scales = "free") +
  geom_histogram()
```

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

## Warning: Removed 24 rows containing non-finite values (stat_bin).
```



We are only interested in the total number of promotions accepted by the customers, since we don't have details about the nature of each promotion.

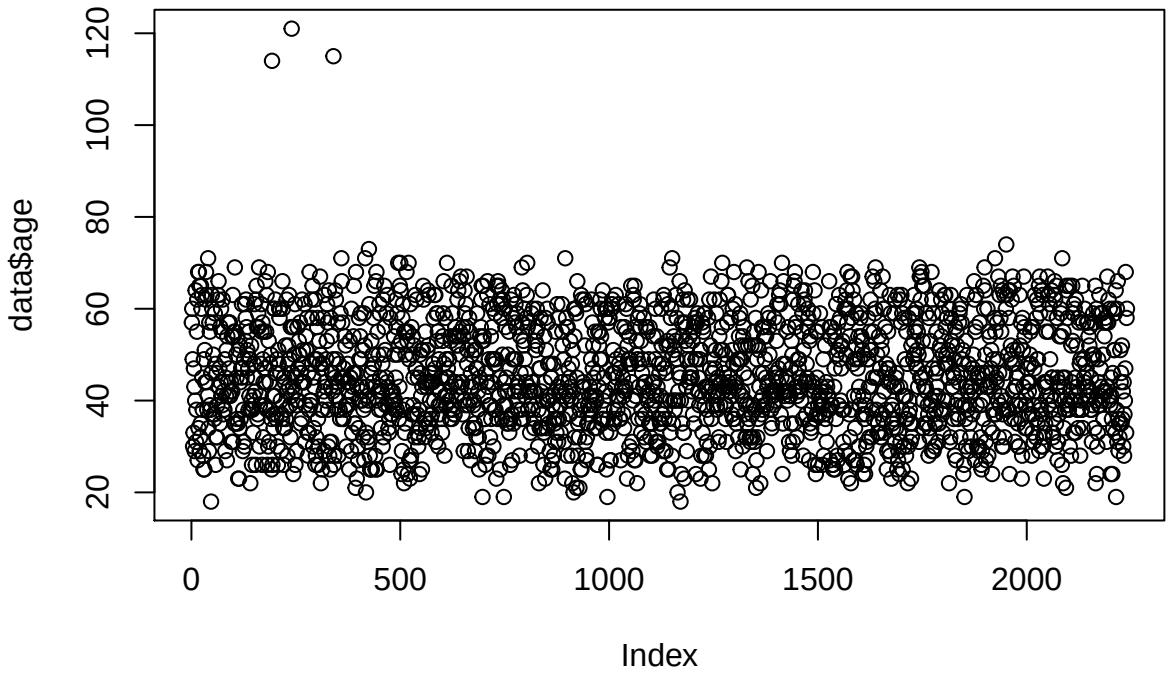
```
data$AcceptedCmpTotal <- data$AcceptedCmp1 + data$AcceptedCmp2 + data$AcceptedCmp3 + data$AcceptedCmp4 +
```

The dataframe contains many variables, some are superfluous for our analysis (web visits and purchases, cumplains, catalog purchases, Z_Revenus and Z_CostContact which we don't have information about)

```
data$Complain <- data$NumWebVisitsMonth <- data$NumWebPurchases <- data$NumCatalogPurchases <- data$Z_R
```

We want to calculate the age of the customers. If we proceed with "2021 - data$Year_Birth$"$ we would get their current age. It makes *data*Year_Birth", although here we are only assuming that it was indeed registered in 2014.

```
data$age <- 2014 - data$Year_Birth
plot(data$age)
```



We see 3 outliers who seems to be older than 110 years old. The corresponding birth years are 1893, 1900 and 1899. The first one could be corrected by 1993, the second one would be due to 2 typing errors which is improbable, and the third could be replaced by 1999 but it corresponds to someone who has a PhD education level, which is unlikely at age 15. Since the dataset is very big, we can choose to delete these lines.

```
which(data$age>110)
```

```
## [1] 193 240 340
```

```
data <- data[-c(193, 240, 340),]
```

Marital_Status can be simplified in only a few levels, and transformed into a factor. Since the "other" section represents less than 1% of the participants, it is not enough to model it as a factor. We then transform some relevant variables into factors.

```
data$Marital_Status <-  factor(data$Marital_Status, labels = c("Other", "Single", "Single", "Married",
data$Marital_Status[data$Marital_Status=="Other"] <- NA; data$Marital_Status = droplevels(data$Marital_

data$Education <- factor(data$Education)
data$Teenhome <-  factor(data$Teenhome)
data$AcceptedCmpTotal <-  factor(data$AcceptedCmpTotal)
data$Kidhome <-  factor(data$Kidhome, labels = c("no", "yes", "yes"))
```

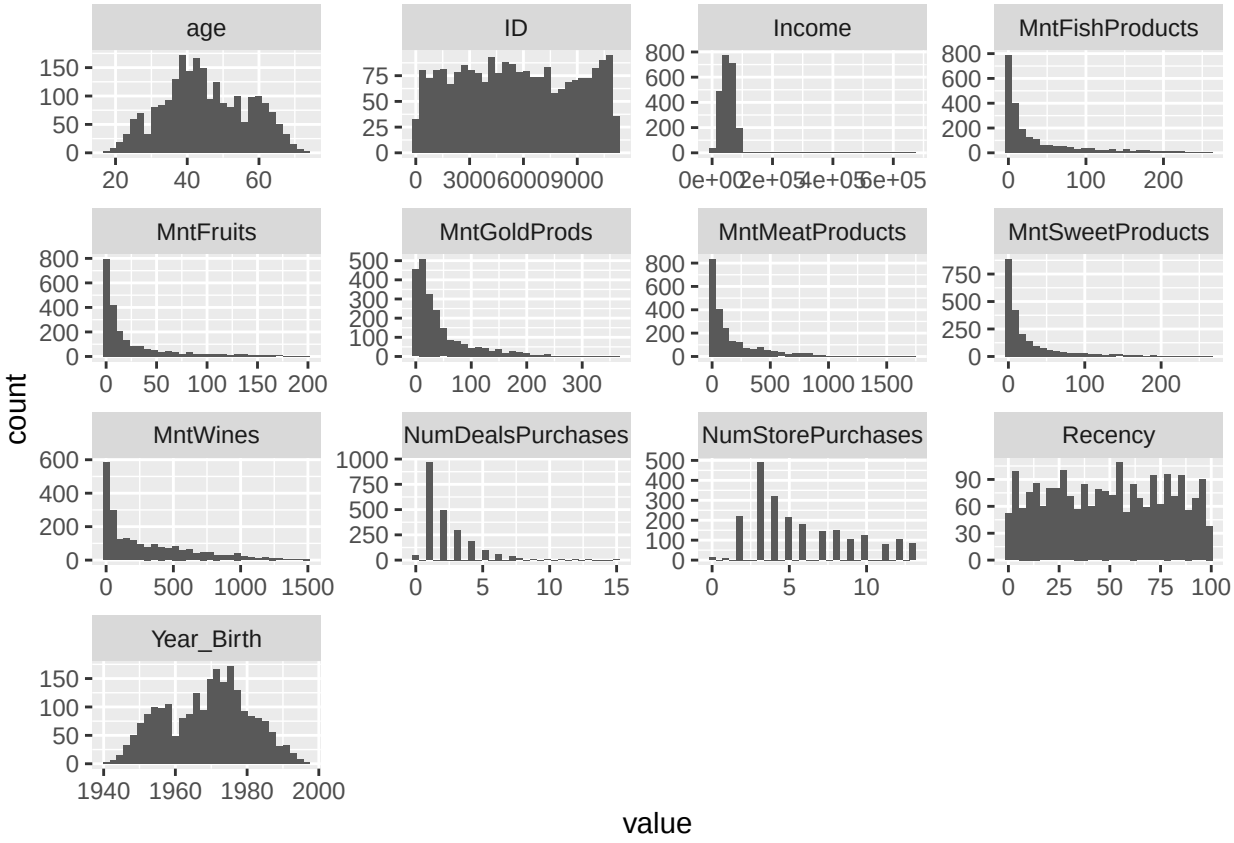
For Kidhome, we fused the answers "1" and "2" because there are only 2% of "2" which is not enough information to model it a one separate factor.

We now want to plot all the variables again, and check again whether anything is abnormal.

```
data %>%
  keep(is.numeric) %>%
  gather() %>%
  ggplot(aes(value)) +
  facet_wrap(~key, scales = "free") +
  geom_histogram()
```

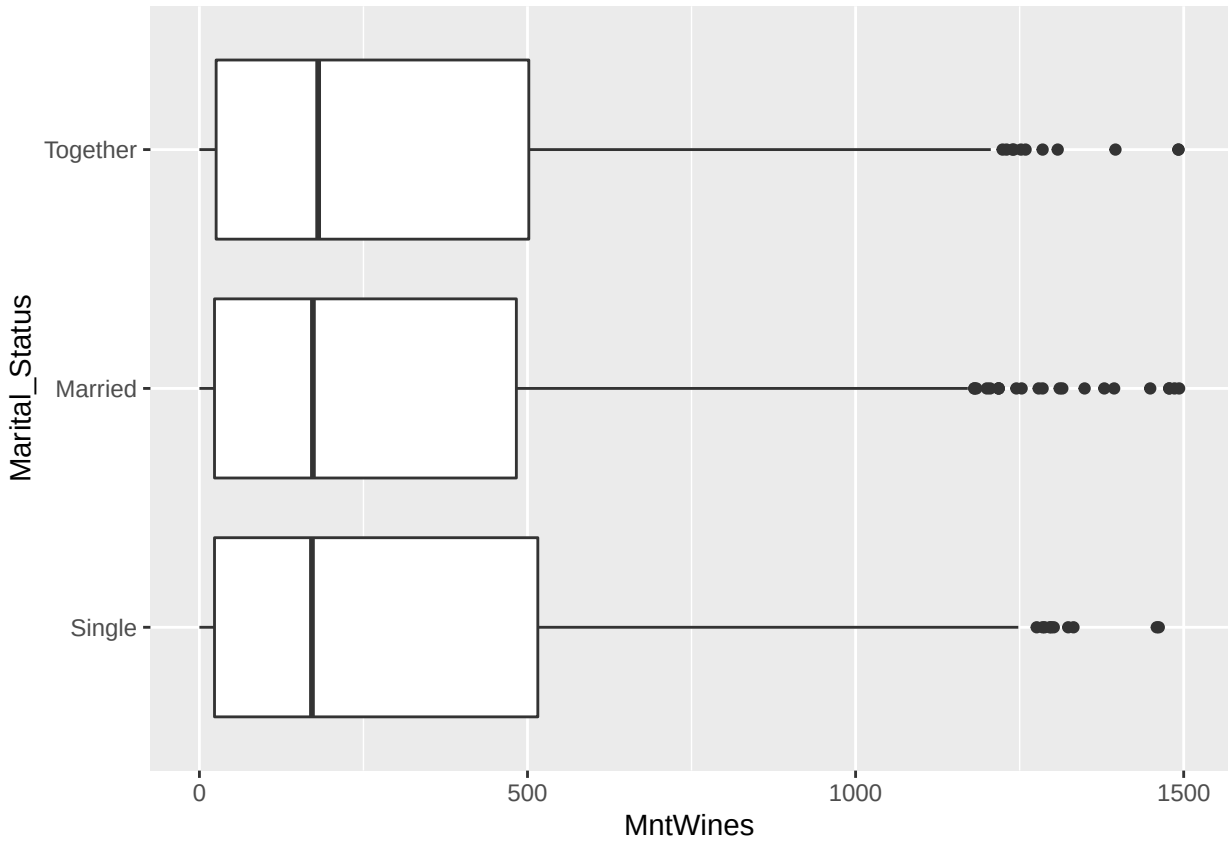
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 24 rows containing non-finite values (stat_bin).
```

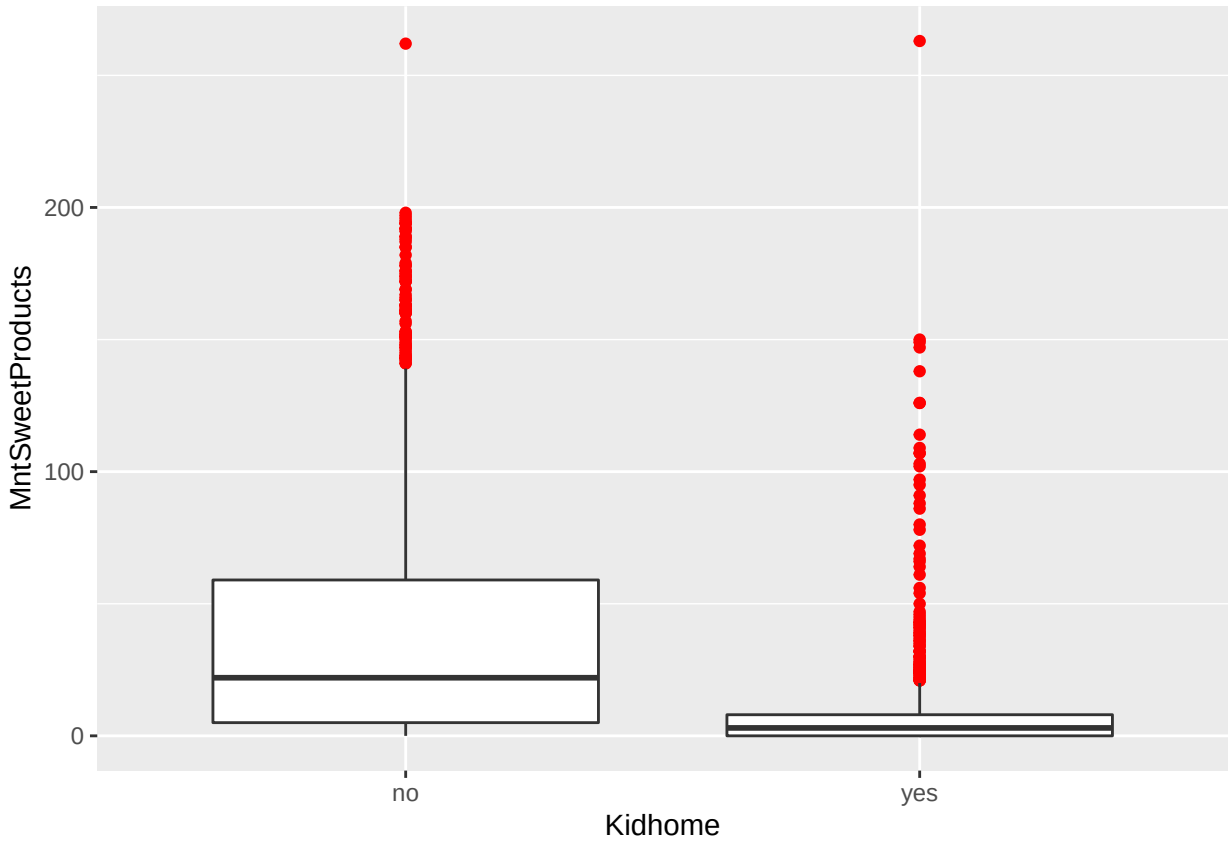


III - Hypotheses

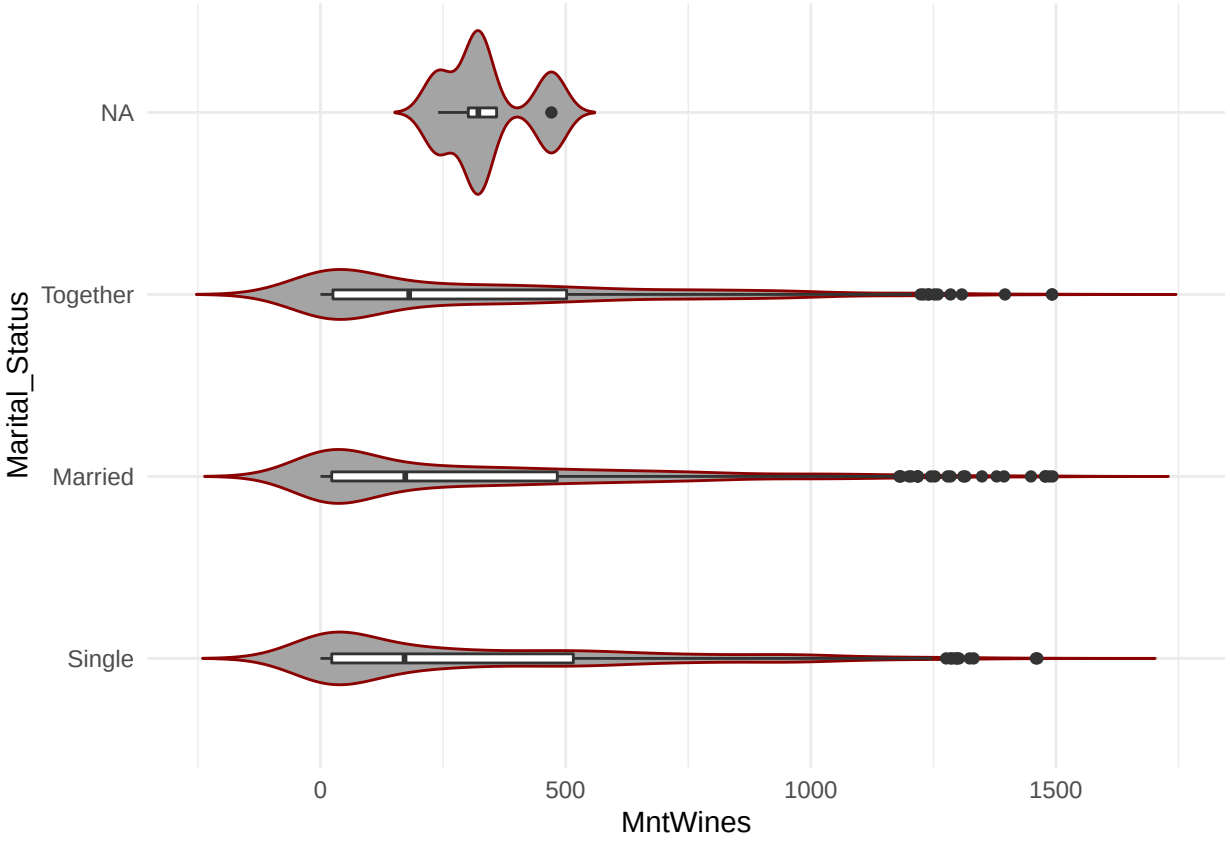
```
#?????
data %>%
      filter(!is.na(Marital_Status)) %>%
  # filter on non-missing values
ggplot(aes(MntWines, Marital_Status)) + geom_boxplot(na.rm = TRUE)
```




```
ggplot(data, aes(Kidhome, MntSweetProducts))+ geom_boxplot(outlier.colour =  "red") #+  geom_point(posi
```



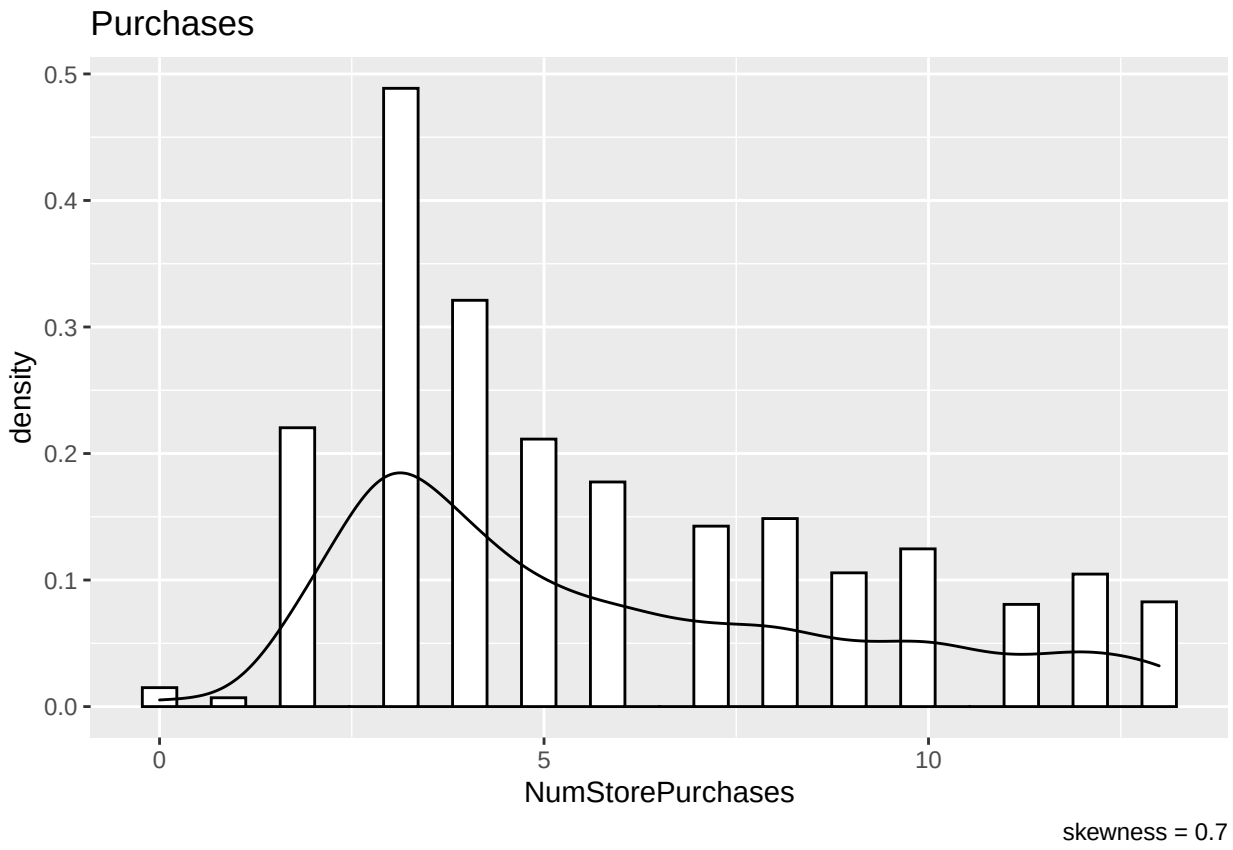

```
ggplot(data, aes(x=MntWines, y=Marital_Status)) +
  geom_violin(trim=FALSE, fill='#A4A4A4', color="darkred")+
  geom_boxplot(width=0.05) + theme_minimal()
```



In order to consider NumStorePurchases and NumDealsPurchases as response variables for linear variables, we first have to check normality.

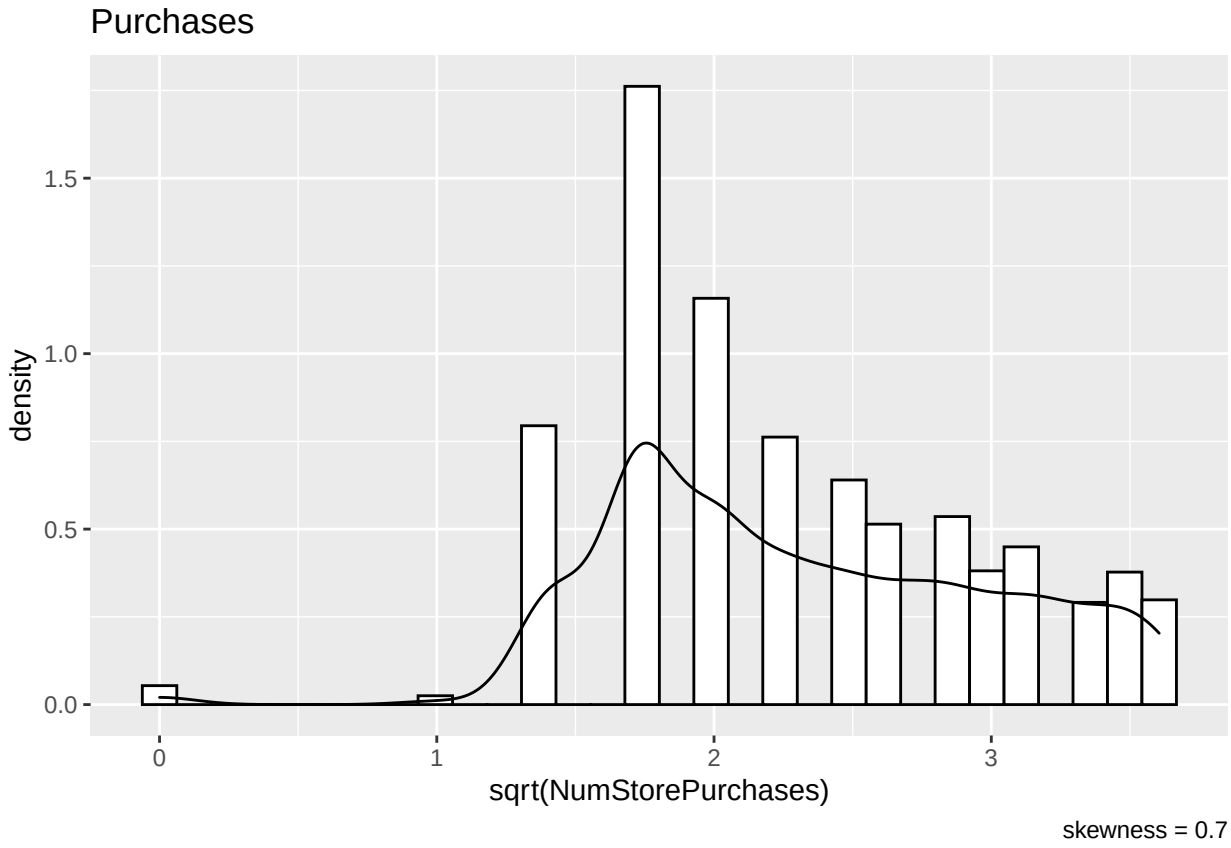
```
ggplot(data, aes(x = NumStorePurchases)) +
  geom_histogram(aes(y = ..density..),
                 colour = 1, fill = "white") +
  geom_density(adjust = 1) + labs(title = "Purchases",
 caption = paste("skewness =", round(moments::skewness(data$NumStorePurchases, na.rm = TRUE),2)))
```

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




```
ggplot(data, aes(x = sqrt(NumStorePurchases))) +
  geom_histogram(aes(y = ..density..),
                 colour = 1, fill = "white") +
  geom_density(adjust = 1) + labs(title = "Purchases",
 caption = paste("skewness =", round(moments::skewness(data$NumStorePurchases, na.rm = TRUE),2)))
```

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

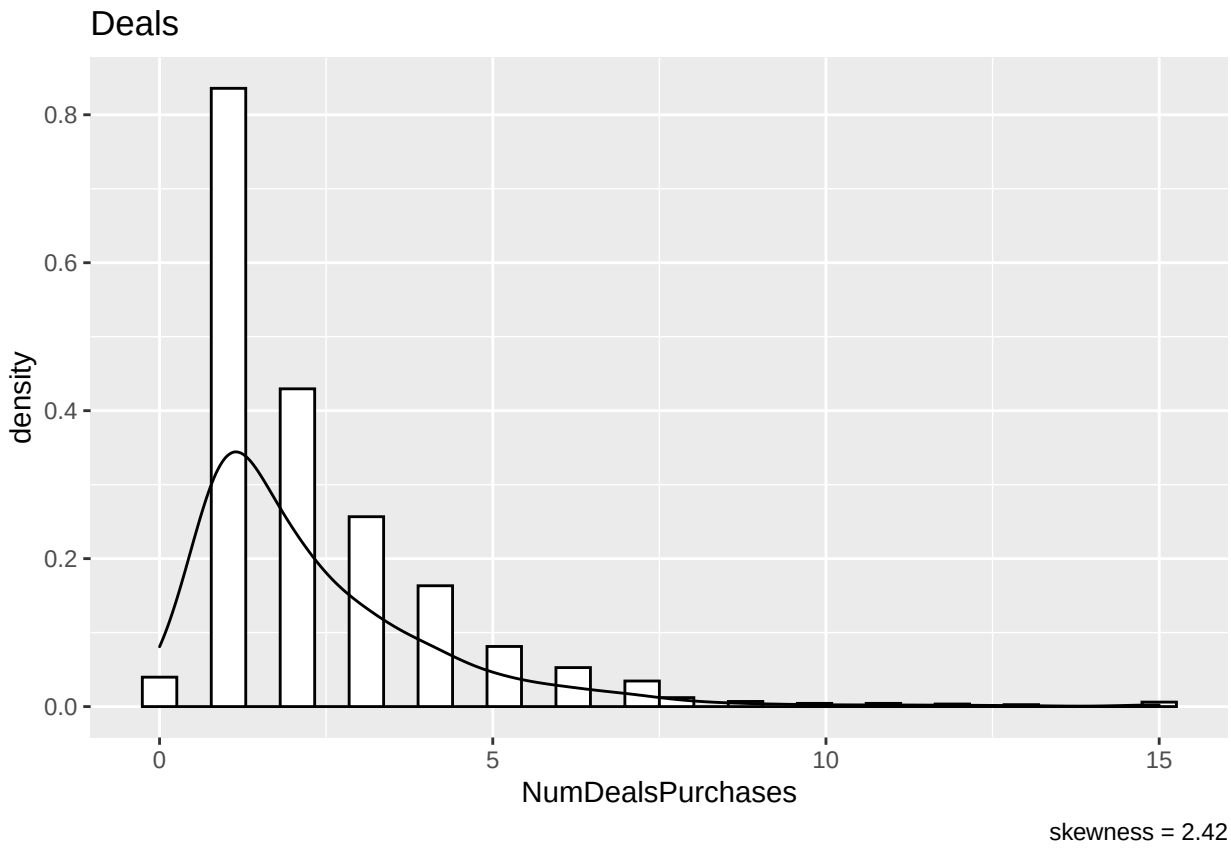



```
#what should we do with NumStorePurchases ? even while using sqrt, there is no normality
```

Here we will check the normality for NumDealsPurchases. It seems like skewness is better if we use the square root formula instead, although it is still not ideal.

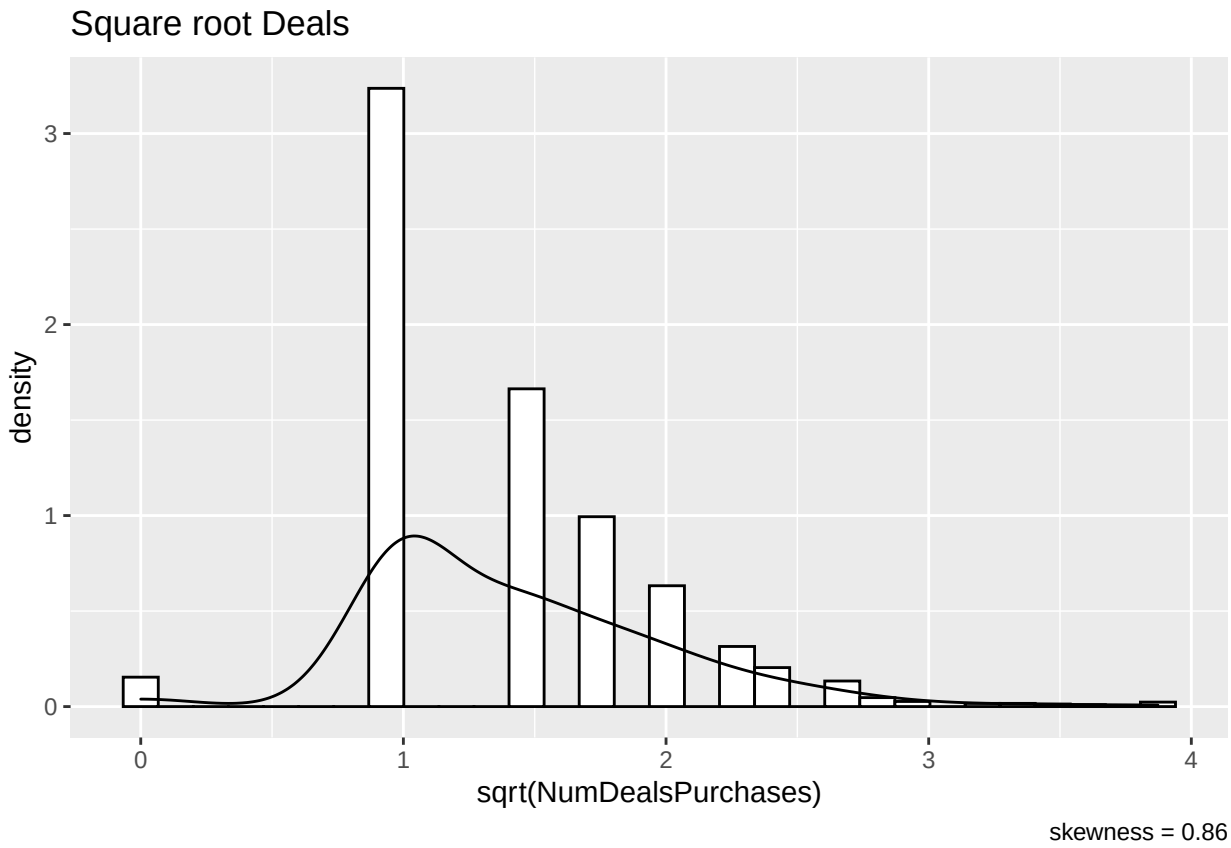
```
ggplot(data, aes(x = NumDealsPurchases)) +
  geom_histogram(aes(y = ..density..),
                 colour = 1, fill = "white") +
  geom_density(adjust = 2) + labs(title = "Deals",
  caption = paste("skewness =", round(moments::skewness(data$NumDealsPurchases, na.rm = TRUE),2)))
```

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




```
ggplot(data, aes(x = sqrt(NumDealsPurchases))) +
  geom_histogram(aes(y = ..density..),
                 colour = 1, fill = "white") +
  geom_density(adjust = 2) + labs(title = "Square root Deals",
  caption = paste("skewness =", round(moments::skewness(sqrt(data$NumDealsPurchases), na.rm = TRUE),2)))
```

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

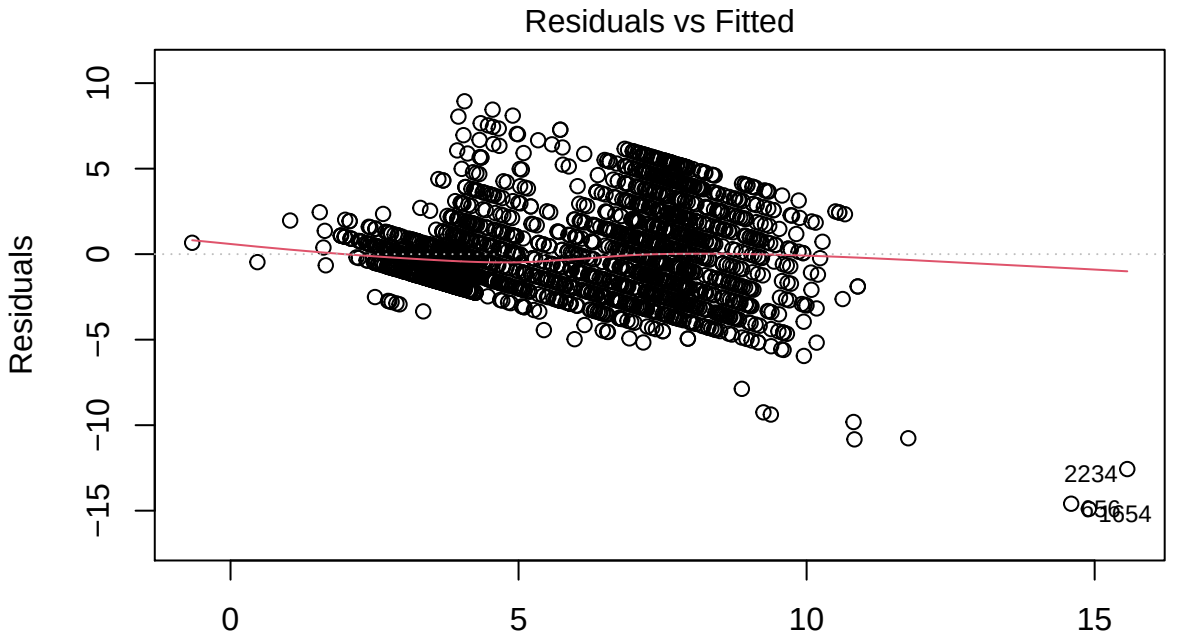


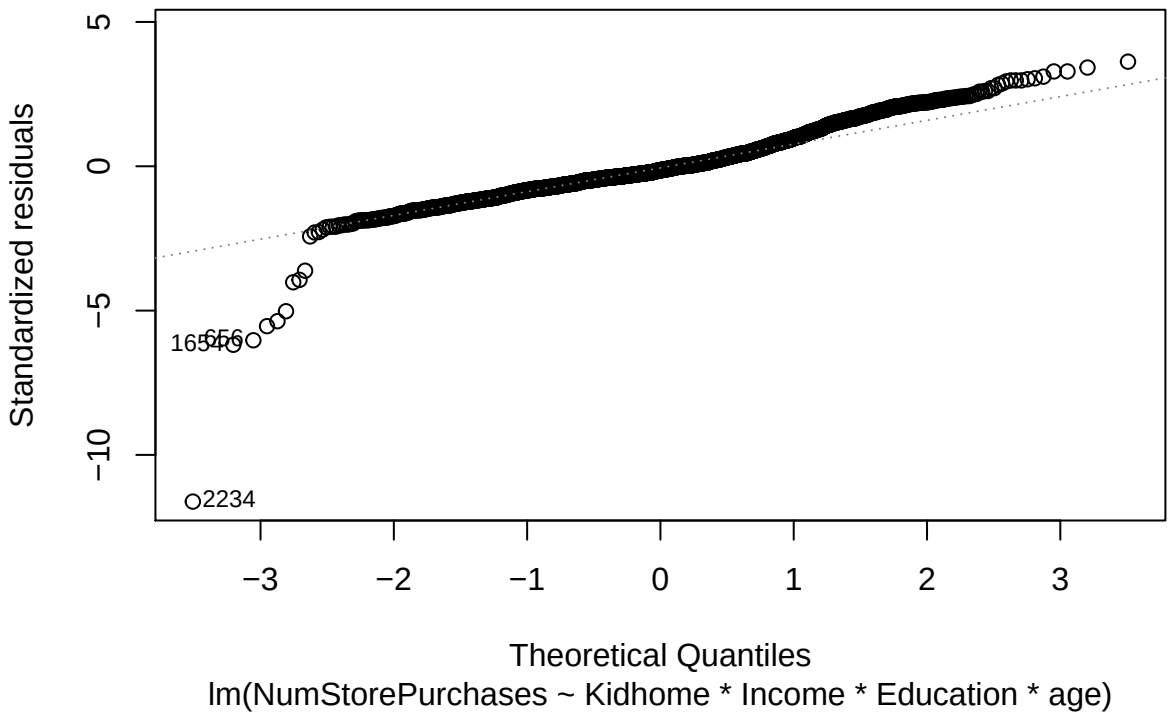
IV - Modelling

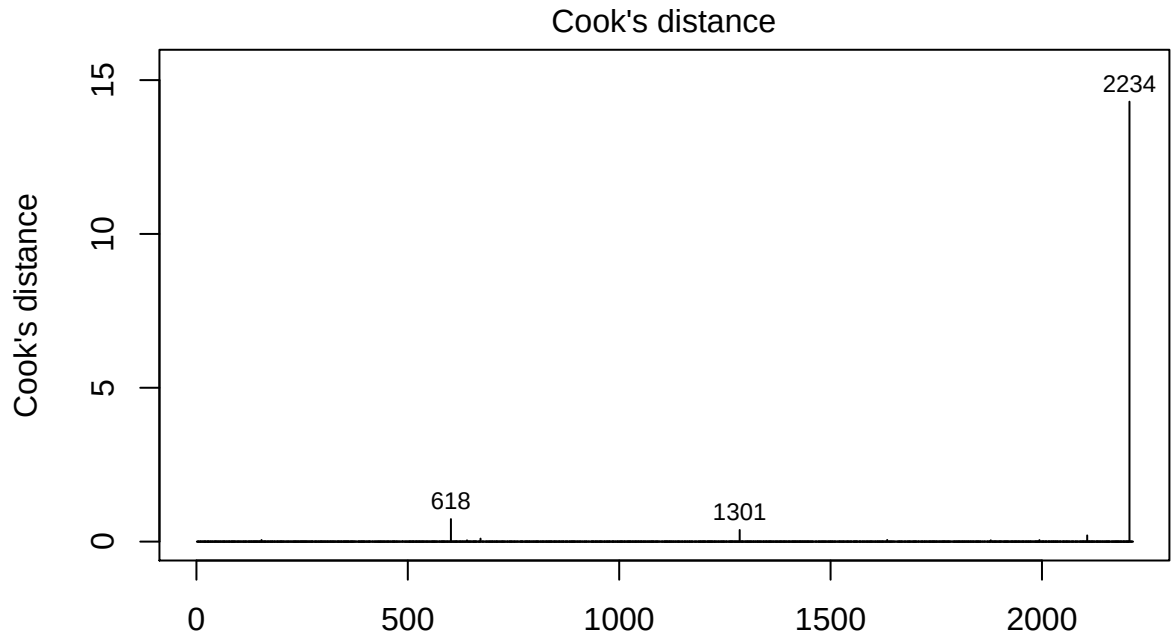
A. Linear modeal

A.1. Creating the model

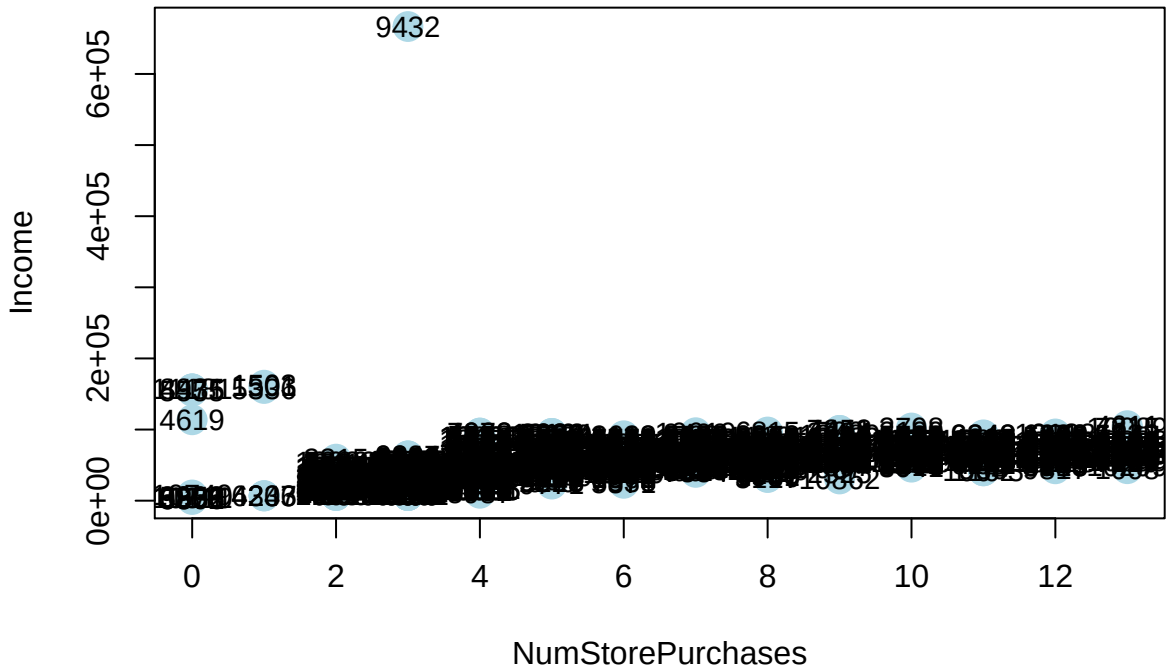
```
m1 <- lm(data=data, NumStorePurchases ~ Kidhome*Income*Education*age)
plot(m1, c(1:2,4), ask=F)
```


Residuals vs Fitted
Residuals
10
5
0
−5
−10
−15
0
5
10
15
2234
656
1654
Fitted values
lm(NumStorePurchases ~ Kidhome * Income * Education * age)


Normal Q−Q
Standardized residuals
5
0
−5
−10
1654
656
2234
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(NumStorePurchases ~ Kidhome * Income * Education * age)



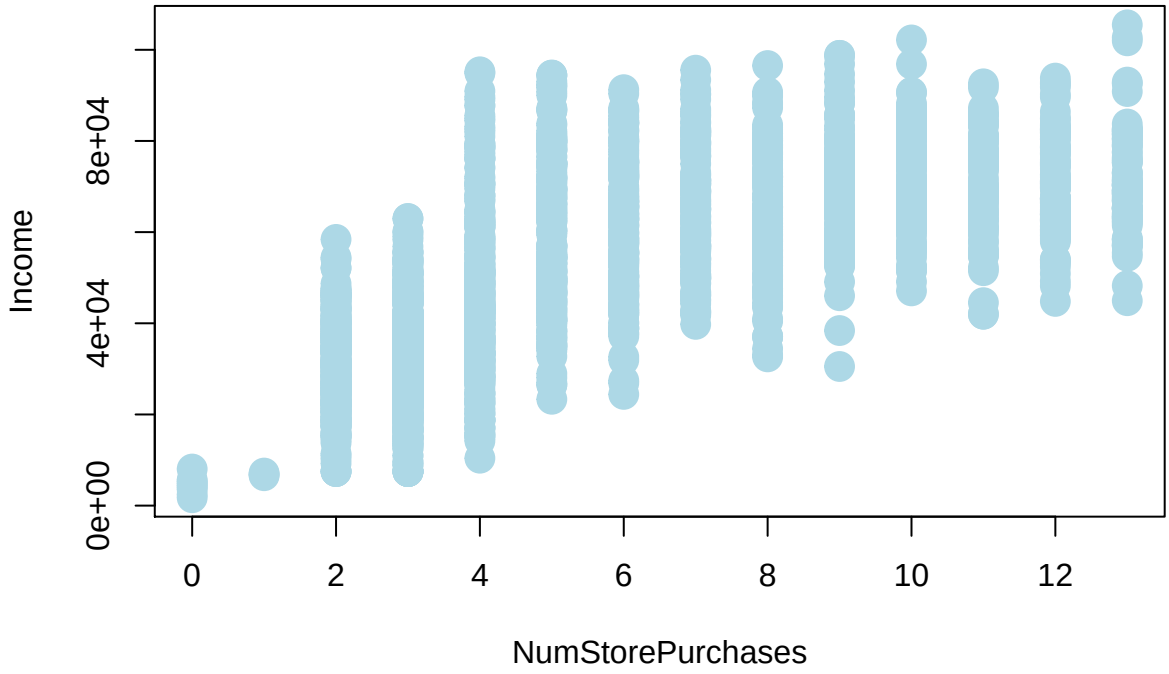

```
plot(Income ~NumStorePurchases, col="lightblue", pch=19, cex=2,data)
text(Income ~NumStorePurchases, labels=ID,data, cex=0.9, font=1)
```



A.2.

Assessing outliers : here we observe peculiar outliers for "Income". One income is equal to 666 666, and when income is higher than 150 000, people probably don't respond. We can safely remove these outliers.

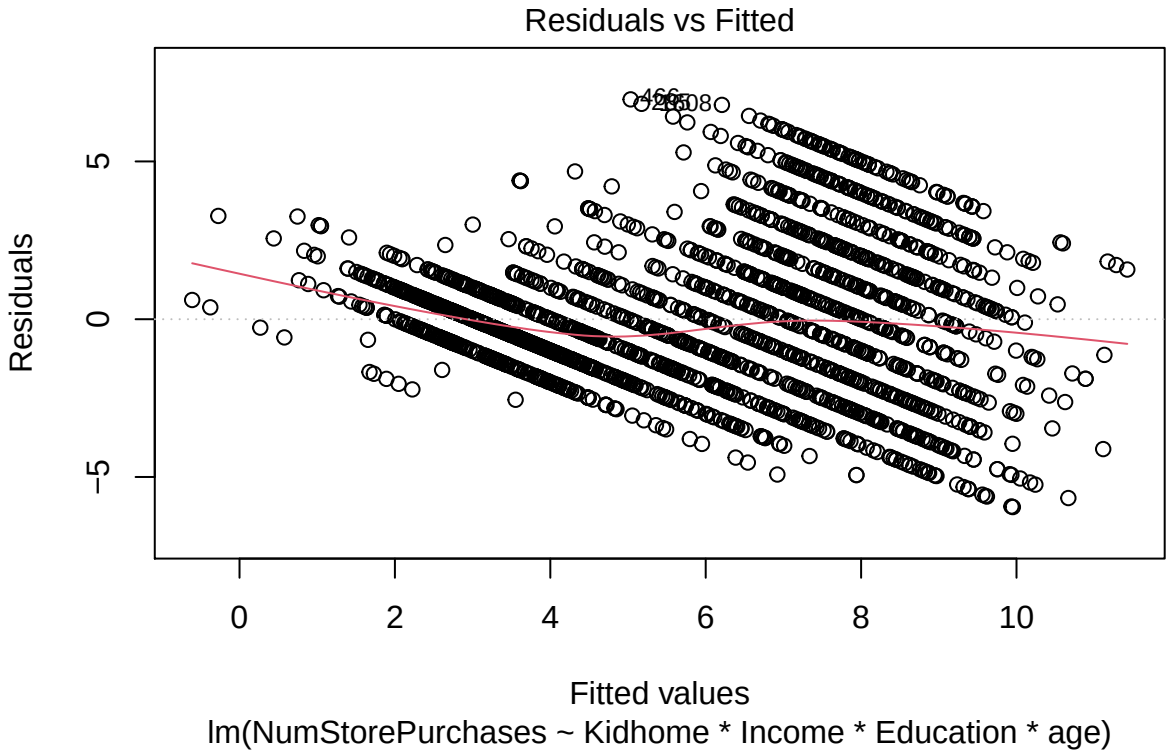
```
newdf = data %>%
  filter(!ID %in% c(9432, 5555, 4619, 5336, 1501, 1503, 8475, 4931, 11181) )

plot(Income ~NumStorePurchases, col="lightblue", pch=19, cex=2,data=newdf)
```

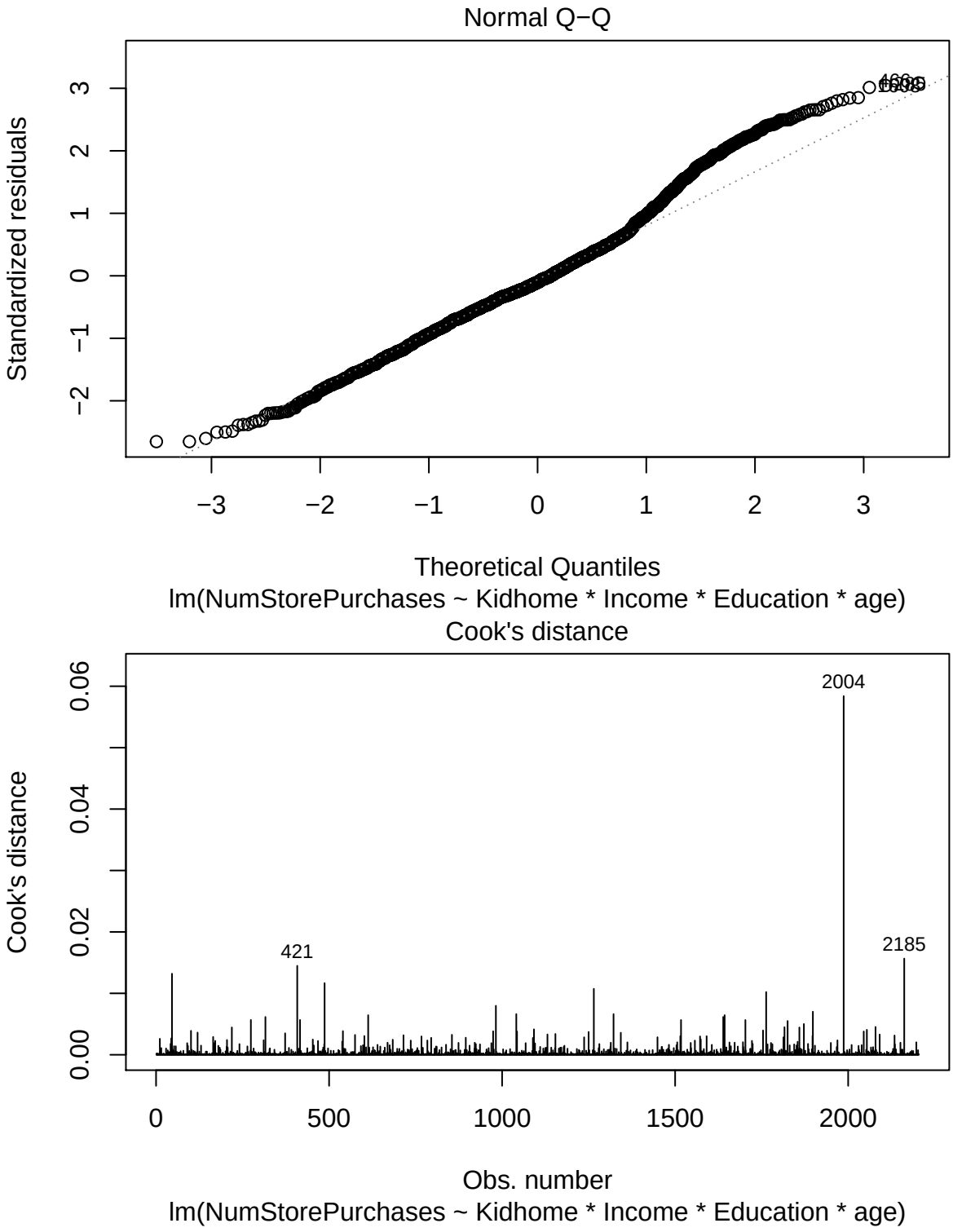


A.3.

Reassessing without the outliers

```
m1 <- lm(data=newdf, NumStorePurchases ~ Kidhome*Income*Education*age)
plot(m1, c(1:2,4), ask=F)
```





A.4.

Filtering the variables to keep : AIC Here we use the stepAIC function to select the model that has the best AIC.

```
ms <- MASS::stepAIC(m1, direction = "both", trace = FALSE) #il choisit le meilleur AIC
ms$anova
```

```
## Stepwise Model Path
```

```
## Analysis of Deviance Table
##
## Initial Model:
## NumStorePurchases ~ Kidhome * Income * Education * age
##
## Final Model:
## NumStorePurchases ~ Kidhome + Income + Education + age + Kidhome:Income +
##     Kidhome:Education + Income:Education + Kidhome:age + Income:age +
##     Education:age + Kidhome:Income:Education + Income:Education:age
##
##
##                              Step Df  Deviance Resid. Df Resid. Dev      AIC
## 1                                                  2164   11138.08 3650.691
## 2 - Kidhome:Income:Education:age  4 26.205697      2168   11164.28 3647.870
## 3        - Kidhome:Education:age  4 14.097325      2172   11178.38 3642.652
## 4           - Kidhome:Income:age  1  8.240739      2173   11186.62 3642.276
```

A.5. Computing the final model

```
finalm1 <- lm(data=newdf,NumStorePurchases ~ Kidhome + Income + Education + age + Kidhome:Income + Kidh
```

A.6. Checking for interferences We first use the effectsize function : everything that has 0.00 on the left of the 90% CI column has a "meaningless" effect size, but we still keep them on the model. We also call the sjPlot function to plot all the estimates or to plot only one term at a time.

```
parameters::model_parameters(anova(finalm1))
```

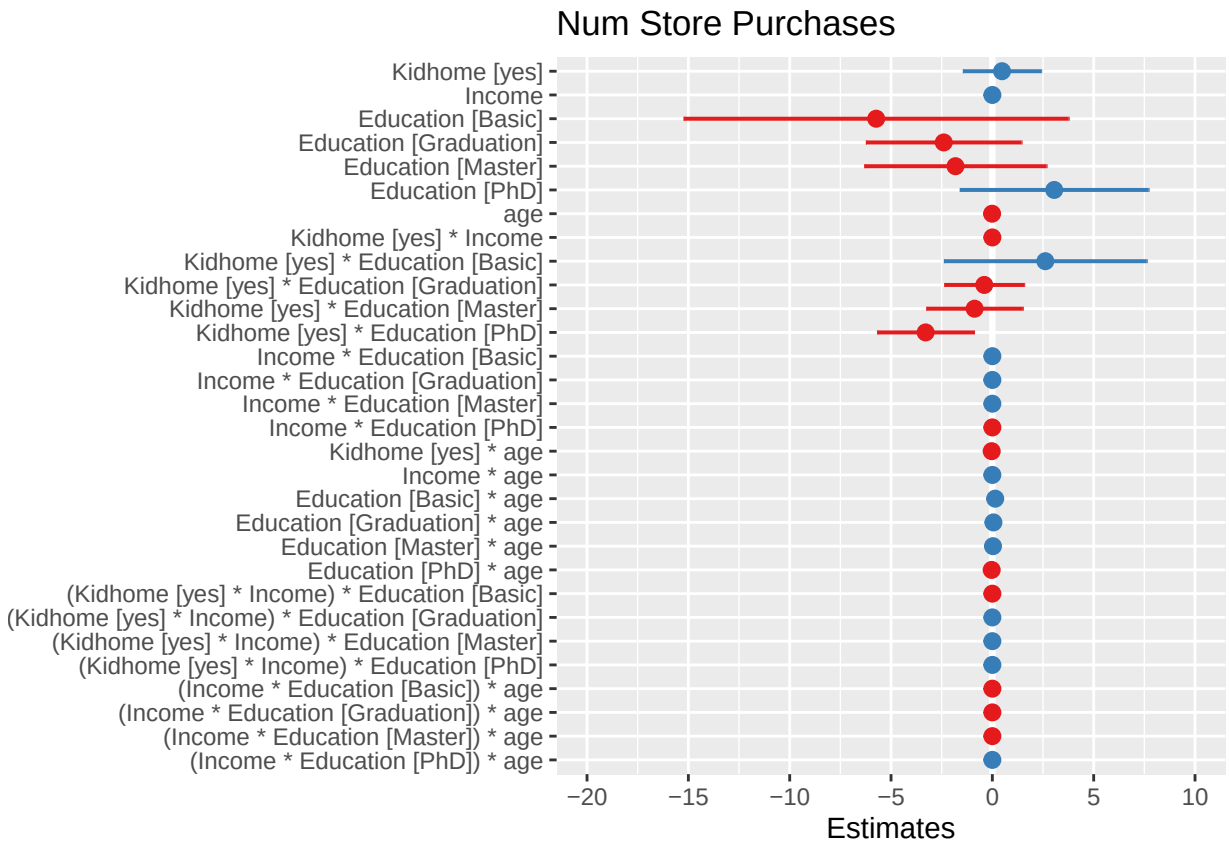
```
## Parameter                | Sum_Squares |   df | Mean_Square |       F |      p
## ------------------------------------------------------------------------------
## Kidhome                  |     6242.08 |    1 |     6242.08 | 1212.52 | < .001
## Income                   |     5446.16 |    1 |     5446.16 | 1057.92 | < .001
## Education                |       10.29 |    4 |        2.57 |    0.50 | 0.736
## age                      |       24.65 |    1 |       24.65 |    4.79 | 0.029
## Kidhome:Income           |        4.87 |    1 |        4.87 |    0.95 | 0.331
## Kidhome:Education        |       31.63 |    4 |        7.91 |    1.54 | 0.189
## Income:Education         |       26.11 |    4 |        6.53 |    1.27 | 0.280
## Kidhome:age              |       36.69 |    1 |       36.69 |    7.13 | 0.008
## Income:age               |        5.97 |    1 |        5.97 |    1.16 | 0.282
## Education:age            |       11.41 |    4 |        2.85 |    0.55 | 0.696
## Kidhome:Income:Education |       52.82 |    4 |       13.20 |    2.56 | 0.037
## Income:Education:age     |       49.16 |    4 |       12.29 |    2.39 | 0.049
## Residuals                |    11186.62 | 2173 |        5.15 |         |
##
## Anova Table (Type 1 tests)
```

```
effectsize::eta_squared(finalm1,ci = 0.9)
```

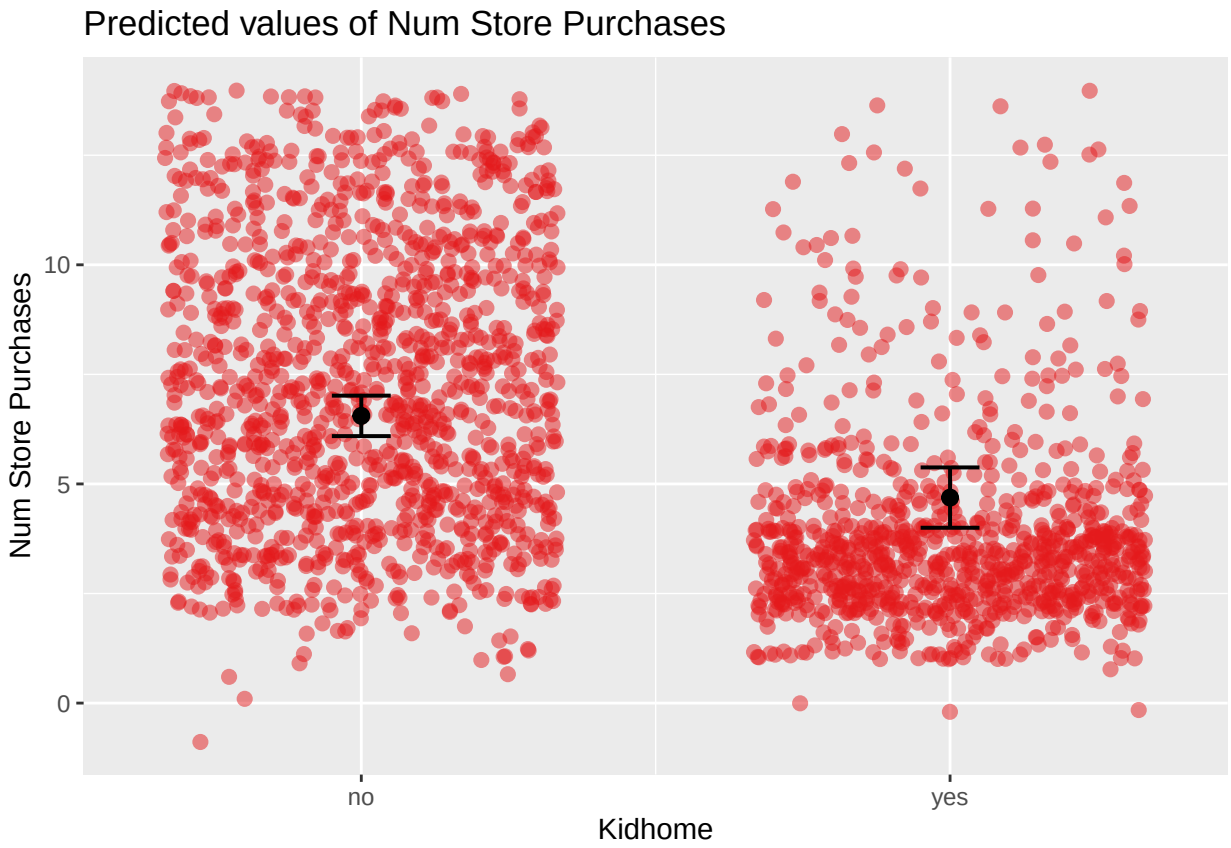
```
## # Effect Size for ANOVA (Type I)
##
## Parameter                | Eta2 (partial) |       90% CI
## ---------------------------------------------------------
## Kidhome                  |           0.36 | [0.34, 1.00]
## Income                   |           0.33 | [0.31, 1.00]
## Education                |       9.19e-04 | [0.00, 1.00]
## age                      |       2.20e-03 | [0.00, 1.00]
## Kidhome:Income           |       4.35e-04 | [0.00, 1.00]
## Kidhome:Education        |       2.82e-03 | [0.00, 1.00]
```

```
## Income:Education          |       2.33e-03 | [0.00, 1.00]
## Kidhome:age               |       3.27e-03 | [0.00, 1.00]
## Income:age                |       5.33e-04 | [0.00, 1.00]
## Education:age             |       1.02e-03 | [0.00, 1.00]
## Kidhome:Income:Education  |       4.70e-03 | [0.00, 1.00]
## Income:Education:age      |       4.38e-03 | [0.00, 1.00]
##
## - One-sided CIs: upper bound fixed at (1).
```

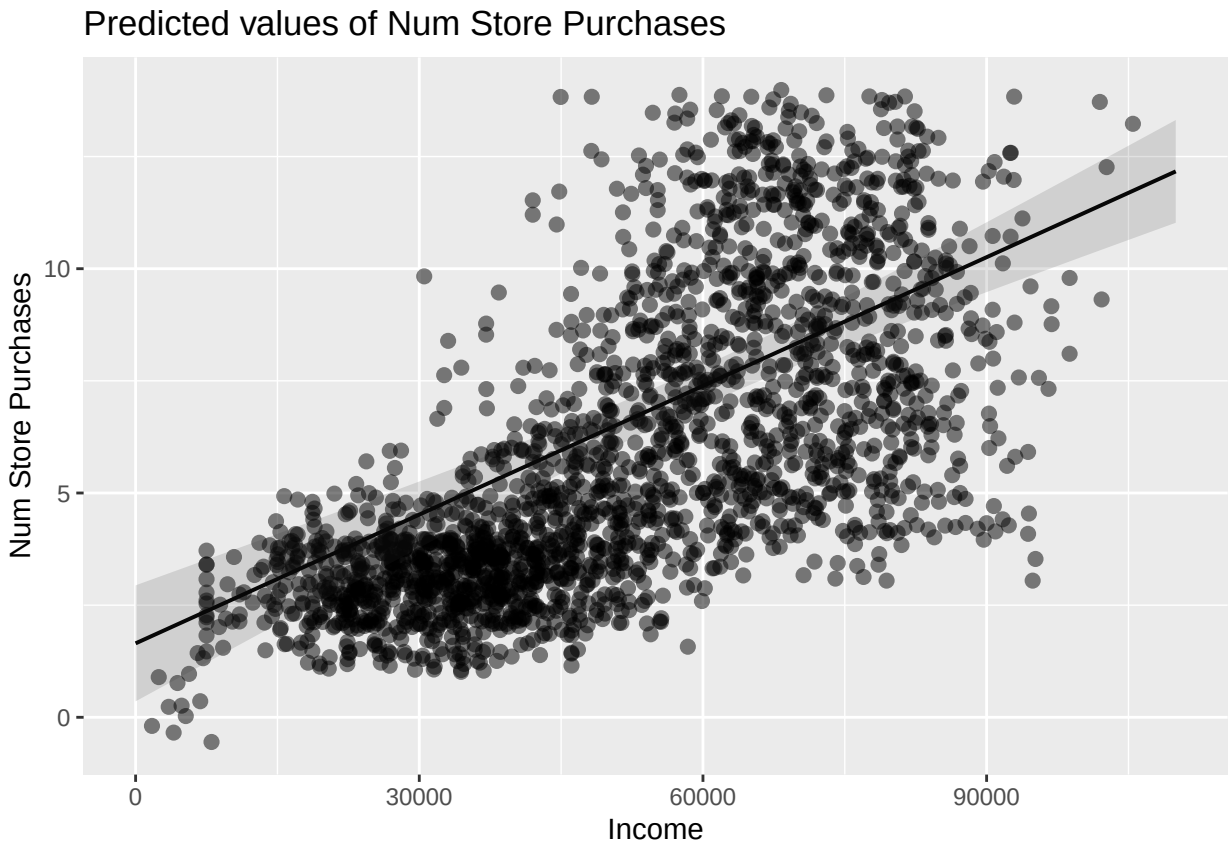
```
sjPlot::plot_model(finalm1)
```




```
sjPlot::plot_model(finalm1, type = "pred", terms = "Kidhome", show.data  = T, jitter = 1)
```




```
sjPlot::plot_model(finalm1, type = "pred", terms = "Income", show.data  = T, jitter = 1)
```

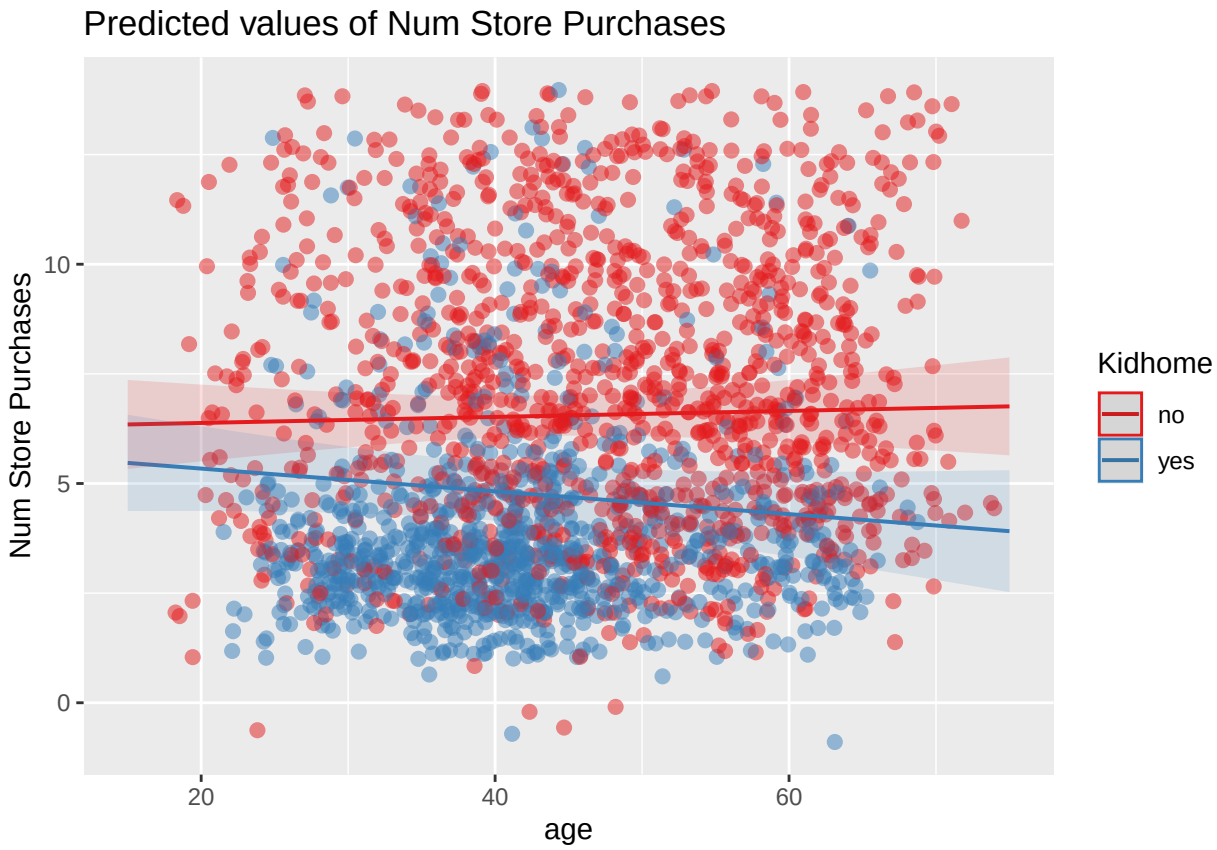



```
sjPlot::plot_model(finalm1, type = "pred", terms = c("Income", "Kidhome"), show.data  = T, jitter = 1)
```


```
sjPlot::plot_model(finalm1, type = "pred", terms = c("Income", "Education"), show.data  = T, jitter =
```


```
sjPlot::plot_model(finalm1, type = "pred", terms = c("Income", "age"), show.data  = T, jitter = 1)
```


```
sjPlot::plot_model(finalm1, type = "pred", terms = c("age", "Kidhome"), show.data  = T, jitter = 1)
```




```
#what is this for?
sjPlot::tab_model(finalm1, rm.terms = c("*Education.Q", "Education^4", "Income:Education.C", "Education
```

Num Store Purchases

Predictors

Estimates

CI

p

(Intercept)

2.25

-1.26 – 5.76

0.208

Kidhome [yes]

0.48

-1.45 – 2.41

0.628

Income

0.00

0.00 – 0.00

0.010

Education [Basic]

-5.73

-15.23 – 3.77

0.237

Education [Graduation]

-2.40

-6.24 – 1.44

0.221

Education [Master]

-1.82

-6.32 – 2.68

0.427

Education [PhD]

3.05

-1.61 – 7.71

0.199

age

-0.01

-0.10 – 0.07

0.750

Kidhome [yes] * Income

-0.00

-0.00 – 0.00

0.383

Kidhome [yes] * Education[Basic]

2.61

-2.39 – 7.62

0.306

Kidhome [yes] * Education[Graduation]

-0.40

-2.37 – 1.57

0.692

Kidhome [yes] * Education[Master]

-0.88

-3.25 – 1.50

0.470

Kidhome [yes] * Education[PhD]

-3.29

-5.68 – -0.91

0.007

Income * Education[Basic]

0.00

-0.00 – 0.00

0.298

Income * Education[Graduation]

0.00

-0.00 – 0.00

0.143

Income * Education[Master]

0.00

-0.00 – 0.00

0.542

Income * Education [PhD]

-0.00

-0.00 – 0.00

0.298

Kidhome [yes] * age

-0.03

-0.05 – -0.01

0.003

Income * age

0.00

-0.00 – 0.00

0.570

Education [Basic] * age

0.14

-0.08 – 0.36

0.210

Education [Graduation] *age

0.05

-0.03 – 0.14

0.234

Education [Master] * age

0.03

-0.07 – 0.13

0.549

Education [PhD] * age

-0.03

-0.14 – 0.07

0.512

(Kidhome [yes] * Income)* Education [Basic]

-0.00

-0.00 – 0.00

0.392

(Kidhome [yes] * Income)* Education [Graduation]

0.00

-0.00 – 0.00

0.374

(Kidhome [yes] * Income)* Education [Master]

0.00

-0.00 – 0.00

0.153

(Kidhome [yes] * Income)* Education [PhD]

0.00

0.00 – 0.00

0.005

(Income * Education[Basic]) * age

-0.00

-0.00 – 0.00

0.209

(Income * Education[Graduation]) * age

-0.00

-0.00 – 0.00

0.113

(Income * Education[Master]) * age

-0.00

-0.00 – 0.00

0.574

(Income * Education[PhD]) * age

0.00

-0.00 – 0.00

0.716

Observations

2204

R2 / R2 adjusted

0.516 / 0.510

B. Principal component analysis

```
#  pm1 <- prcomp(data,~ + "Year_Birth"  +  "Education"  +  "Marital_Status"  +  "Income" + #"Kidhome"
clean_data <-data[rowSums(is.na(data))==0, ]
pm1<-prcomp(clean_data[,-c(1,3,4,6,7,8,18)],  scale=TRUE)
summary(pm1)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     2.0965 1.4242 1.04714 1.00079 0.90502 0.80789 0.70029
## Proportion of Variance 0.3663 0.1690 0.09138 0.08346 0.06826 0.05439 0.04087
## Cumulative Proportion  0.3663 0.5353 0.62667 0.71013 0.77839 0.83278 0.87364
##                           PC8     PC9    PC10    PC11      PC12
## Standard deviation     0.65473 0.64171 0.61837 0.54168 6.022e-15
## Proportion of Variance 0.03572 0.03432 0.03187 0.02445 0.000e+00
## Cumulative Proportion  0.90937 0.94368 0.97555 1.00000 1.000e+00
```

```
names(data)
```

```
##  [1] "ID"                "Year_Birth"        "Education"
##  [4] "Marital_Status"    "Income"            "Kidhome"
##  [7] "Teenhome"          "Dt_Customer"       "Recency"
## [10] "MntWines"          "MntFruits"         "MntMeatProducts"
## [13] "MntFishProducts"   "MntSweetProducts"  "MntGoldProds"
## [16] "NumDealsPurchases" "NumStorePurchases" "AcceptedCmpTotal"
## [19] "age"
```

#scale= standardise tout, car certains var. ont des échelles très différentes