Activitat 4

Marc Cervera Rosell

21-06-2024

```
setRepositories(ind=2) # Per descarregar paquets de CRAN
```

1 Preprocessament

1.1 Variables Income i Year_Birth

```
tryCatch({
  data <- read.csv("marketing.csv", header = TRUE)
  print("Fitxer llegit correctament")
}, error = function(e){
  cat("ERROR en la lectura del fitxer:",conditionMessage(e),"\n")
})
```

```
## [1] "Fitxer llegit correctament"
```

```
columns <- names(data)
type <- sapply(data, class)
for (i in seq_along(columns)) {
  cat("La columna", columns[i], "es de tipus", type[i], "\n")
}
```

```
## La columna ID es de tipus integer
## La columna Year_Birth es de tipus integer
## La columna Education es de tipus character
## La columna Marital_Status es de tipus character
## La columna Income es de tipus integer
## La columna Kidhome es de tipus integer
## La columna Teenhome es de tipus integer
## La columna Dt_Customer es de tipus character
## La columna Recency es de tipus integer
## La columna MntWines es de tipus integer
## La columna MntFruits es de tipus integer
## La columna MntMeatProducts es de tipus integer
## La columna MntFishProducts es de tipus integer
## La columna MntSweetProducts es de tipus integer
## La columna MntGoldProds es de tipus integer
## La columna NumDealsPurchases es de tipus integer
## La columna NumWebPurchases es de tipus integer
## La columna NumCatalogPurchases es de tipus integer
## La columna NumStorePurchases es de tipus integer
## La columna NumWebVisitsMonth es de tipus integer
## La columna AcceptedCmp3 es de tipus integer
## La columna AcceptedCmp4 es de tipus integer
```

```
## La columna AcceptedCmp5 es de tipus integer
## La columna AcceptedCmp1 es de tipus integer
## La columna AcceptedCmp2 es de tipus integer
## La columna Complain es de tipus integer
## La columna Z_CostContact es de tipus integer
## La columna Z_Revenue es de tipus integer
## La columna Response es de tipus integer
```

Després d'observar els tipus de les variables del conjunt de dades, solament es procedirà a fer un canvi de tipus. Aquest canvi es produirà en la variable *Dt_Customer* que passarà de ser de tipus *character* a tipus *date*.

```
data_transformed <- transform(data,
                              Dt_Customer = as.Date(Dt_Customer))
```

```
columns <- names(data_transformed)
type <- sapply(data_transformed, class)
for (i in seq_along(columns)) {
  cat("La columna", columns[i], "es de tipus", type[i], "\n")
}
```

```
## La columna ID es de tipus integer
## La columna Year_Birth es de tipus integer
## La columna Education es de tipus character
## La columna Marital_Status es de tipus character
## La columna Income es de tipus integer
## La columna Kidhome es de tipus integer
## La columna Teenhome es de tipus integer
## La columna Dt_Customer es de tipus Date
## La columna Recency es de tipus integer
## La columna MntWines es de tipus integer
## La columna MntFruits es de tipus integer
## La columna MntMeatProducts es de tipus integer
## La columna MntFishProducts es de tipus integer
## La columna MntSweetProducts es de tipus integer
## La columna MntGoldProds es de tipus integer
## La columna NumDealsPurchases es de tipus integer
## La columna NumWebPurchases es de tipus integer
## La columna NumCatalogPurchases es de tipus integer
## La columna NumStorePurchases es de tipus integer
## La columna NumWebVisitsMonth es de tipus integer
## La columna AcceptedCmp3 es de tipus integer
## La columna AcceptedCmp4 es de tipus integer
## La columna AcceptedCmp5 es de tipus integer
## La columna AcceptedCmp1 es de tipus integer
## La columna AcceptedCmp2 es de tipus integer
## La columna Complain es de tipus integer
## La columna Z_CostContact es de tipus integer
## La columna Z_Revenue es de tipus integer
## La columna Response es de tipus integer
```

S'observa que després de l'aplicació de la funció *transform()* el tipus de la variable *Dt_Customer* queda modificat.

Finalment, cal excloure les variables *Z_CostContact* i *Z_Revenue*, atès que són variables de control i que així s'indica a l'enunciat de l'activitat.

```
columns_to_exclude <- c("Z_CostContact", "Z_Revenue")
data_with_no_control_variables <- data_transformed[, !(names(data_transformed) %in%
                                                         columns_to_exclude)]
print(names(data_with_no_control_variables))
```

```
##  [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"  "NumWebPurchases"     "NumCatalogPurchases"
## [19] "NumStorePurchases"  "NumWebVisitsMonth"   "AcceptedCmp3"
## [22] "AcceptedCmp4"       "AcceptedCmp5"        "AcceptedCmp1"
## [25] "AcceptedCmp2"       "Complain"            "Response"
```

S'observa que en treure per pantalla les columnes del *dataset data_with_no_control_variables* les variables de control indicades anteriorment ja no hi són. Per tant, aquest conjunt final queda completament operatiu per a poder treballar.

1.2 Valors absents

Abans de res, encara que l'enunciat ja ho diu, és bona pràctica comprovar si realment hi ha valors absents.

```
any(is.na(data_with_no_control_variables$Income))
```

```
## [1] TRUE
```

```
any(is.na(data_with_no_control_variables$Year_Birth))
```

```
## [1] FALSE
```

```
library(VIM)
```

```
## Loading required package: colorspace
## Loading required package: grid
## VIM is ready to use.
## Suggestions and bug-reports can be submitted at: https://github.com/statistikat/VIM/issues
##
## Attaching package: 'VIM'
## The following object is masked from 'package:datasets':
##
##     sleep
```

Primer cal seleccionar aquelles variables que s'usaran per al càlcul de la distància de Gower. Com s'especifica en l'enunciat seran les variables 10 a 15.

```
variables_gower <- c("MntWines", "MntFruits", "MntMeatProducts", "MntFishProducts",
                     "MntSweetProducts", "MntGoldProds")
```

Com s'indica en les instruccions de l'activitat en desenvolupament, per imputar els valors *NA* de la varible *Income* s'aplica la funció *kNN* de la llibreria *VIM*.

```
income_imputed <- kNN(data_with_no_control_variables, variable = "Income",
                      dist_var = variables_gower, k = 5)
```

Un cop aplicada la funció *kNN* s'observa que la variable *Income* ja no té valors *NA*

```
any(is.na(income_imputed$Income))
```

```
## [1] FALSE
```

En primer lloc, abans de calcular la mitjana d'edat cal seleccionar les persones que són vídues.

```
widowed_people <- subset(income_imputed, Marital_Status == "Widow")
```

Un cop seleccionades aquestes persones ja es pot procedir al càlcul de la mitjana d'edat.

```
mean_widow_age <- mean(widowed_people$Year_Birth)
cat("L'any de naixement mitjà de les persones vidues es:"
    , round(mean_widow_age))
```

```
## L'any de naixement mitjà de les persones vidues es: 1959
```

```
income_imputed$Year_Birth <- ifelse(is.na(income_imputed$Year_Birth), mean_widow_age,
                                    income_imputed$Year_Birth)
```

```
any(is.na(income_imputed$Year_Birth))
```

```
## [1] FALSE
```

S'observa que després de l'execució de la cel · la que conté la sentència *ifelse* els valors *NA* de la variable *Year_Birth* ja no són tals.

```
any(is.na(income_imputed))
```

```
## [1] FALSE
```

Un cop eliminats els valors *NA* de les variables *Income* i *Year_Birth* es pot observar (en la cel · la anterior) el conjunt de dades queda completament lliure de valors *NA*, per tant, cal redenominar aquest conjunt a *markclean*.

```
markclean <- income_imputed
```

Finalment, es demana una reflexió sobre el nombre de valors *NA* del fitxer, per tant, cal, en primer lloc, comptar aquests valors.

```
na_per_column <- colSums(is.na(data_with_no_control_variables))
total_na_values <- sum(na_per_column)
cat("El nombre total de valors NA es:", total_na_values)
```

```
## El nombre total de valors NA es: 24
```

Tot i que el nombre de valors *NA* del fitxer no és considerablement elevat, és important abordar aquests valors. El fet de l'existència de valors *NA* pot ser un indicador de problemes en les dades com: problemes en el moment de la recollida de les dades o senzillament que en el moment d'introduir les dades en el fitxer s'ha comès error humà. Com s'acaba d'esmentar, malgrat que el nombre de valors absents no és molt significatiu, és molt important gestionar aquests buits per evitar problemes en les anàlisis que es puguin arribar a fer tant en aquesta activitat com les anàlisis que pugui fer una altra persona completament aliena a la UOC.

2 Estadística descriptiva

2.1 Income

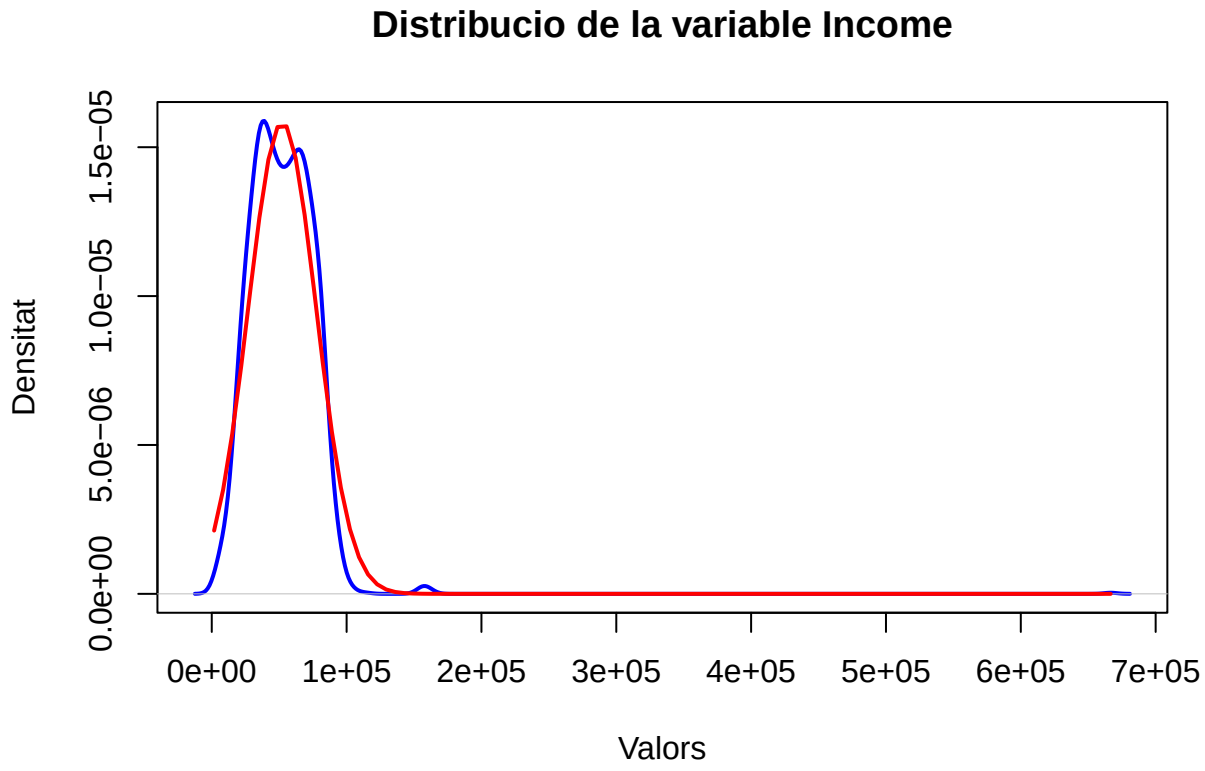
```
mean_abs <- abs(mean(markclean$Income))
standard_deviation <- sd(markclean$Income)
coeffitient_of_variation <- standard_deviation / mean_abs
cat("El coeficient de variacio de la variable Income es:", coeffitient_of_variation)
```

```
## El coeficient de variacio de la variable Income es: 0.4824468
```

Es pot observar un coeficient de variació de 0.4824468. Això indica que, efectivament, existeix una variabilitat en la distribució, però que no és excessiva. Per tant, l'ingrés mitjà sí que és un valor representatiu de la distribució dels ingressos.

Per respondre a la segona pregunta cal, prèviament, utilitzar alguna eina per visualitzar la distribució de les dades de la variable. En aquest cas s'ha escollit veure la distribució en un gràfic de densitat. També s'introduirà una campana de Gauss (densitat normal) de les dades d'estudi per veure com de lluny estan de distribuir-se de manera normal.

```
plot(density(markclean$Income),
     main = "Distribucio de la variable Income",
     xlab = "Valors",
     ylab = "Densitat",
     col = "blue",
     lwd = 2)
values_normal_distribution <- seq(min(markclean$Income), max(markclean$Income),
                                  length = 100)
normal_distribution <- dnorm(values_normal_distribution, mean = mean_abs,
                             sd = standard_deviation) # Campana de Gauss de les dades
lines(values_normal_distribution, normal_distribution, col = "red", lwd = 2)
```



Després d'observar ambdós gràfics, es pot observar que les dades no normalitzades (línia blava) no segueixen una distribució normal atès que la forma de la corba en el gràfic de densitat queda molt allunyada de la forma de la línia vermella que representa les dades normalitzades.

2.2 Education

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

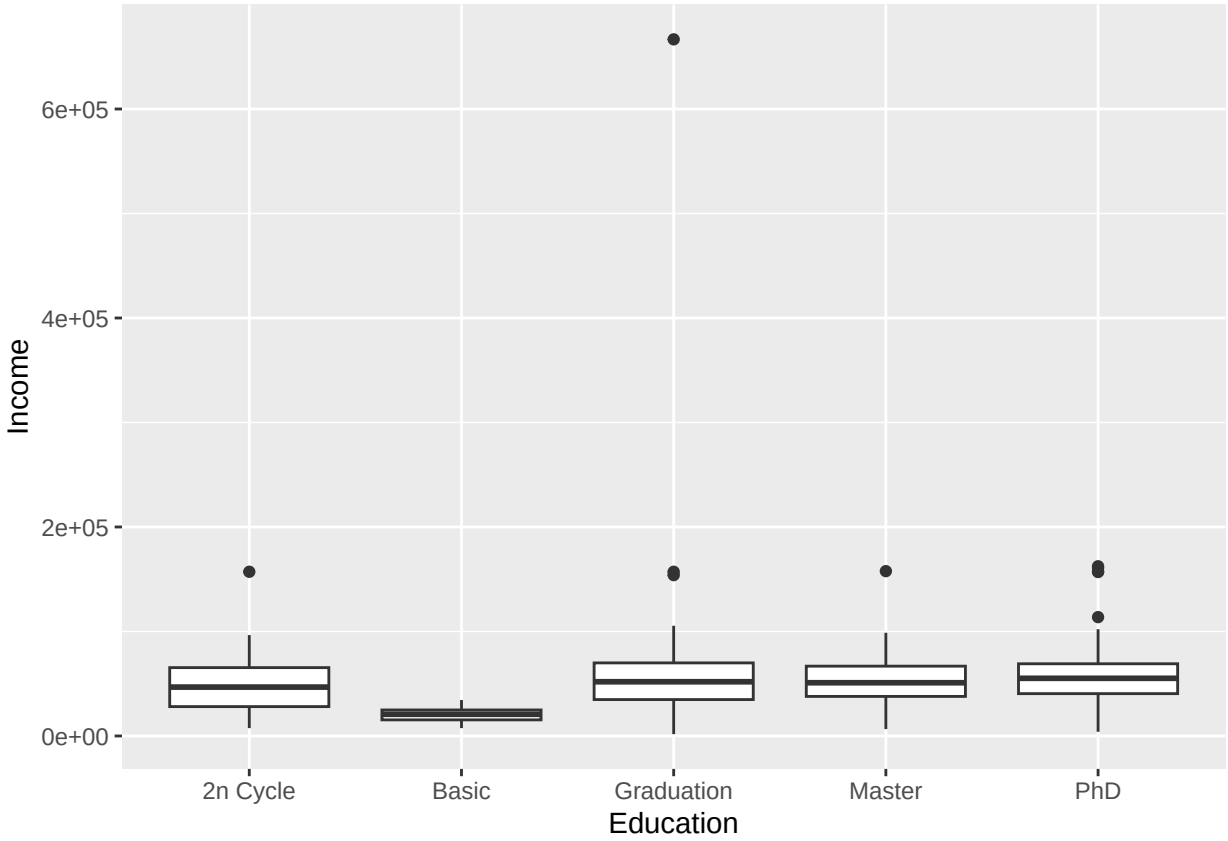
```
library(ggplot2)
library(magrittr) # Per l'operador %>%
```

L'operador "%>%" és un operador que crea un *pipe* i permet que el resultat d'una funció passi com a primer argument de la següent funció.

```
markclean %>%
  group_by(Education) %>%
  summarize(
    mean_value = mean(Income),
    observations = n(),
    deviation = sd(Income)
  ) %>%
  arrange(desc(Education))
```

```
## # A tibble: 5 x 4
##   Education  mean_value observations deviation
##   <chr>           <dbl>        <int>     <dbl>
## 1 PhD            56061.          486    20544.
## 2 Master         53014.          370    20082.
## 3 Graduation     52616.         1127    28125.
## 4 Basic          20306.           54     6235.
## 5 2n Cycle       48051.          203    23298.
```

```
markclean %>%
  arrange(desc(Education)) %>%
  ggplot(aes(x = Education, y = Income)) +
  geom_boxplot() +
  labs(x = "Education", y = "Income", tittle = "Ingressos segons nivell educatiu")
```



En primer lloc, es pot observar que la posició de la mediana (línia interna de la caixa), es troba just al centre en els nivells educatius "Graduation", "Master" i "PhD", per tant, en aquests tres casos es pot concloure que el 50% dels valors estan per sota de la mediana i l'altre 50% per sobre. Per als nivells educatius "2n Cycle" i "Basic", es pot observar que en el cas del primer nivell educatiu comentat, la línia de la mediana es troba lleugerament desplaçada a la part superior de la caixa i en el cas del nivell "Basic" la línia de la mediana està a la part superior de la caixa, per tant, en aquests dos casos les dades presenten una asimetria positiva cosa que indica que la majoria de les dades es troben a l'esquerra de la mediana. El fet de tenir una asimetria positiva és indicatiu, en aquest cas, que una petita proporció de persones tenen més ingressos que la majoria de les persones del mateix nivell educatiu.

Observant els bigotis de les caixes, es pot veure que n'hi ha de més curt i de més llargs. Segons la longitud dels bigotis de cada caixa, es podrà veure com d'agrupades estan les dades, és a dir, es podrà observar com són de dispersos els valors extrems. Per interpretar els bigotis de les caixes cal mirar la seva longitud. Com més llargs siguin els bigotis més dispersos estaran els valors extrems, és a dir, els valors extrems estaran més lluny de la resta de valors. Per contra, com més curts siguin els bigotis de les caixes, menys dispersos estaran els valors extrems i, per tant, més propers estaran a la resta de valors.

Finalment, s'observa la presència de valors atípics. Els valors atípics són els punts que estan situats fora dels bigotis, però això no significa que no siguin valors importants.

3. Estadística inferencial

3.1 Contrast d'hipòtesi per a la diferència de les mitjanes

3.1.1 Escriviu la hipòtesi nul · la i l'alternativa

Hipòtesi nul · la: Els ingressos mitjans de les persones sense estudis universitaris són iguals als de les persones amb estudis universitaris.

Hipòtesi alternativa: Els ingressos mitjans de les persones sense estudis són inferiors als de les persones amb estudis universitaris.

3.1.2 Justificació del test a aplicar

Per a poder aplicar un test de diferència de mitjanes, és condició necessària que les mostres a examinar siguin independents.

Per comprovar la igualtat, o diferència, de les variàncies s'aplicarà un test de Fisher.

Hipòtesi nul · la: Les variàncies són iguals.

Hipòtesi nul · la: Les variàncies són diferents.

```
alpha <- 0.01
# Ingressos no universitaris
income_no_uni <- markclean$Income[markclean$Education %in% c("2n Cycle", "Basic")]
# Ingressos universitaris
income_uni <- markclean$Income[markclean$Education %in% c("Graduation",
                                                          "Master", "PhD")]
mean_no_uni <- mean(income_no_uni)
mean_uni <- mean(income_uni)
n_no_uni <- length(income_no_uni)
n_uni <- length(income_uni)
s_no_uni <- sd(income_no_uni)
s_uni <- sd(income_uni)
c(mean_no_uni, mean_uni, s_no_uni, s_uni, n_no_uni, n_uni)
```

```
## [1] 42221.33 53534.81 23761.62 25096.78   257.00  1983.00
```

```
fobs <- s_no_uni^2 / s_uni^2
fcritL <- qf(alpha/2, df1 = n_no_uni - 1, df2 = n_uni - 2)
fcritU <- qf(1 - alpha/2, df1 = n_no_uni - 1, df2 = n_uni - 2)
pvalue <- min(pf(fobs, df1 = n_no_uni - 1, df2 = n_uni - 2, lower.tail = FALSE),
              pf(fobs, df1 = n_no_uni - 1, df2 = n_uni - 2)) * 2
c(fobs, fcritL, fcritU, pvalue)
```

```
## [1] 0.8964297 0.7769809 1.2620087 0.2613326
```

Atès que el valor observat es troba dins dels límits L i U i que el valor P és major al valor alfa es pot concloure que no hi ha evidència suficient per a rebutjar la hipòtesi nul · la i que amb un 99% de confiança les variàncies d'ambdues poblacions són iguals.

3.1.3 Càlculs

```
combined_s <- sqrt((((n_no_uni - 1) * s_no_uni^2) + ((n_uni - 1) * s_uni^2))/
                   (n_no_uni + n_uni - 2))
t_obs <- (mean_no_uni - mean_uni) / (combined_s * sqrt((1 / n_no_uni) + (1 / n_uni)))
tcritL <- qt(alpha / 2, n_no_uni + n_uni - 2)
tcritU <- qt(1 - alpha / 2, n_no_uni + n_uni - 2)
```

```r
pvalue_test <- pt(abs(t_obs), df = n_no_uni + n_uni - 2, lower.tail = FALSE) * 2
c(t_obs, tcritL, tcritU, pvalue_test)
```

```
## [1] -6.840222e+00 -2.578028e+00  2.578028e+00  1.016752e-11
```

3.1.4 Interpretació del test

Vist que el valor observat (t_obs) es troba fora del rang de valors crítics i que el valor p (pvalue_test) és inferior al valor alfa (0.01) es rebutja la hipòtesi nul · la. Per tant, es pot afirmar que amb un 99% de confiança els ingressos mitjans de les persones sense estudis són inferiors als de les persones amb estudis universitaris.

4. Model de regressió

4.1 Regressió lineal múltiple

```r
markclean$Education <- factor(markclean$Education,
                              levels = c("Basic","2n Cycle","Graduation", "Master", "PhD"))
model <- lm(Income ~ Year_Birth + Kidhome + Teenhome + Education + MntWines + MntFruits +
              MntMeatProducts + MntFishProducts + MntSweetProducts + MntGoldProds +
              NumDealsPurchases + NumCatalogPurchases + NumStorePurchases +
              NumWebVisitsMonth, data = markclean)
summary_model <- summary(model)
print(summary_model)
```

```
##
## Call:
## lm(formula = Income ~ Year_Birth + Kidhome + Teenhome + Education +
##     MntWines + MntFruits + MntMeatProducts + MntFishProducts +
##     MntSweetProducts + MntGoldProds + NumDealsPurchases + NumCatalogPurchases +
##     NumStorePurchases + NumWebVisitsMonth, data = markclean)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -98693  -6278   -301   5379 631265
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         103538.343  66696.226   1.552 0.120712
## Year_Birth             -34.414     33.787  -1.019 0.308532
## Kidhome               1143.139    910.809   1.255 0.209582
## Teenhome              5345.988    832.522   6.421 1.65e-10 ***
## Education2n Cycle    11935.765   2661.163   4.485 7.66e-06 ***
## EducationGraduation  12787.871   2447.197   5.226 1.90e-07 ***
## EducationMaster      12671.850   2570.625   4.929 8.85e-07 ***
## EducationPhD         13944.712   2553.400   5.461 5.26e-08 ***
## MntWines                18.096      1.700  10.645  < 2e-16 ***
## MntFruits               22.053     12.682   1.739 0.082200 .
## MntMeatProducts         21.889      2.695   8.124 7.43e-16 ***
## MntFishProducts          8.715      9.612   0.907 0.364651
## MntSweetProducts        27.746     11.979   2.316 0.020632 *
## MntGoldProds            -2.359      8.369  -0.282 0.778061
## NumDealsPurchases     -166.768    237.553  -0.702 0.482738
## NumCatalogPurchases    686.393    212.981   3.223 0.001288 **
```

```
## NumStorePurchases      553.961    167.452   3.308 0.000954 ***
## NumWebVisitsMonth    -2774.672    210.764 -13.165  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17140 on 2222 degrees of freedom
## Multiple R-squared:  0.5407, Adjusted R-squared:  0.5372
## F-statistic: 153.9 on 17 and 2222 DF,  p-value: < 2.2e-16
```

Per comprovar la qualitat de l'ajust, s'observaran i interpretaran tres valors que atorga el *summary* del model. El primer valor és el *Multiple R-squared*. Aquest valor és de 0.5407 (54.07%), això indica que les variables incloses en el model poden explicar fins al 54.07% de les variacions dels ingressos. El fet que aquest valor superi el 50% és indicatiu de què el model és relativament bo. Per considerar el model bo i no relativament bo, caldria que aquest valor superés el 70%.

El segon valor a interpretar és el *Adjusted R-squared*. Aquest valor és una versió modificada de l'anterior que ha estat ajustat al nombre de predictors del model. Aquest valor és de 0.5372 (53.72%), per tant, el model encara pot explicar el 53.72% de les variacions dels ingressos. Vist que continua superant el 50%, es pot concloure que l'ajust continua sent relativament bo.

Finalment, l'últim valor a considerar per la qualitat de l'ajust és el *Residual standard error* (RSE). Aquest valor indica la dispersió dels punts al voltant de la línia del millor ajustament.

Com el RMSE està calculat en els ingressos (variable depenent), es pot interpretar el resultat del RMSE comparant amb el valor mitjà dels ingressos.

```
mean_income <- mean(markclean$Income)
cat("Ingressos mitjans:", mean_income)
```

```
## Ingressos mitjans: 52236.79
```

S'observa que el RMSE és bastant menor que els ingressos mitjans. En vista d'aquest fet, es conclou que el model realitza relativament bé les prediccions.

4.1.1 Multicolinealitat

```
library(car)
```

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

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

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

```
vif_vals <- vif(model)
```

```
print(vif_vals)
```

```
##                       GVIF Df GVIF^(1/(2*Df))
## Year_Birth        1.248981  1        1.117578
## Kidhome           1.831899  1        1.353477
## Teenhome          1.565626  1        1.251250
## Education         1.235570  4        1.026794
## MntWines          2.494539  1        1.579411
## MntFruits         1.938338  1        1.392242
## MntMeatProducts   2.817909  1        1.678663
```

```
## MntFishProducts     2.100263  1        1.449228
## MntSweetProducts    1.862707  1        1.364810
## MntGoldProds        1.452199  1        1.205072
## NumDealsPurchases   1.605020  1        1.266894
## NumCatalogPurchases 2.952622  1        1.718319
## NumStorePurchases   2.257591  1        1.502528
## NumWebVisitsMonth   1.992712  1        1.411635
```

Després d'observar els resultats de la funció *vif()* és pot observar que la multicol · linealitat del model no és un problema (no significa que no hi hagi multicol · linealitat) atès que cap valor GVIF és superior a 5. Pel que fa als resultats de la tercera columna, els valors tampoc són alarmants tot i que cal estar atent.