Practica 2 tipologia

Ruben Herrera

Practica 2

```
#llamamos a todas las bibliotecas que usaremos
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

## Warning: package 'readr' was built under R version 3.6.2

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(data.table)
```

```
##
## Attaching package: 'data.table'

## The following objects are masked from 'package:dplyr':
##
##     between, first, last

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

```
library (naniar)
```

```
## Warning: package 'naniar' was built under R version 3.6.3
```

```
library(dplyr)
library(gtools)
library(normtest)
library(nortest)
library(plyr)
```

```
## ------------------------------------------------------------------------
```

```
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
```

```
## ------------------------------------------------------------------------------
```

```
##
## Attaching package: 'plyr'
```

```
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

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

```
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 3.6.3
```

```
##
## Attaching package: 'gridExtra'
```

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

1 Descripcion del dataset

Hemos escogido este dataset que consiste en las ventas de sillitas de niños, que hemos encontrado en Kiggle con ello pretendemos saber cual es la variable que mejor predice las ventas, por lo que combinaremos las variables para hacer un estudio de que variables afectan mas a la variacion de ventas, tambien miraremos si hay diferencia de ventas en estados unidos y fuera, asi como en zonas urbanas y rurales vemos que en este caso las muestras estan muy limpios pero he encontrado un dataset que adjuntamos en a carpeta de el mismo tipo de muestras que tiene errores y que vamos a tratar

```
#data set descargado que esta limpio
chair_clean <- read.csv(file="Carseats_training.csv", sep = "," ,dec = ".", stringsAsFactors = FALSE)
#leemos el data set y lo guardamos en chair_clean
head(chair_clean, 20)
```

```
##    ID Sales CompPrice Income Advertising Population Price ShelveLoc Age
## 1   1 10.48       138     72           0        148    94    Medium  27
## 2   2 10.43        77     69           0         25    24    Medium  50
## 3   3  5.32       118     74           6        426   102    Medium  80
## 4   4  7.67       129    117           8        400   101       Bad  36
## 5   5  5.32       152    116           0        170   160    Medium  39
## 6   6 14.37        95    106           0        256    53      Good  52
```

```
## 7   7  4.95       121     41           5        412   110    Medium  54
## 8   8  9.39       117    118          14        445   120    Medium  32
## 9   9  8.80       145     53           0        507   119    Medium  41
## 10 10  4.68       124     46           0        199   135    Medium  52
## 11 11  2.67       115     54           0        406   128    Medium  42
## 12 12  7.41       162     26          12        368   159    Medium  40
## 13 13 11.48       121    120          13        140    87    Medium  56
## 14 14  9.35        98    117           0         76    68    Medium  63
## 15 15  5.71       121     42           4        188   118    Medium  54
## 16 16  4.69       132    113           0        131   124    Medium  76
## 17 17  9.14       134     67           0        286    90       Bad  41
## 18 18  9.32       119     60           0        372    70       Bad  30
## 19 19  8.21       127     44          13        160   123      Good  63
## 20 20  9.44       131     47           7         90   118    Medium  47
##    Education Urban  US
## 1         17   Yes Yes
## 2         18   Yes  No
## 3         18   Yes Yes
## 4         10   Yes Yes
## 5         16   Yes  No
## 6         17   Yes  No
## 7         10   Yes Yes
## 8         15   Yes Yes
## 9         12   Yes  No
## 10        14    No  No
## 11        17   Yes Yes
## 12        18   Yes Yes
## 13        11   Yes Yes
## 14        10   Yes  No
## 15        15   Yes Yes
## 16        17    No Yes
## 17        13   Yes  No
## 18        18    No  No
## 19        18   Yes Yes
## 20        12   Yes Yes
```

```
# vemos una representacion de las 20 primeras filas
str(chair_clean)#vemos un resumen de las caracteriticas de cada variable
```

```
## 'data.frame':    320 obs. of  12 variables:
##  $ ID         : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Sales      : num  10.48 10.43 5.32 7.67 5.32 ...
##  $ CompPrice  : int  138 77 118 129 152 95 121 117 145 124 ...
##  $ Income     : int  72 69 74 117 116 106 41 118 53 46 ...
##  $ Advertising: int  0 0 6 8 0 0 5 14 0 0 ...
##  $ Population : int  148 25 426 400 170 256 412 445 507 199 ...
##  $ Price      : int  94 24 102 101 160 53 110 120 119 135 ...
##  $ ShelveLoc  : chr  "Medium" "Medium" "Medium" "Bad" ...
##  $ Age        : int  27 50 80 36 39 52 54 32 41 52 ...
##  $ Education  : int  17 18 18 10 16 17 10 15 12 14 ...
##  $ Urban      : chr  "Yes" "Yes" "Yes" "Yes" ...
##  $ US         : chr  "Yes" "No" "Yes" "Yes" ...
```

```r
any(is.na(chair_clean)) #vemos que no hay ningun dato NA
```

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

```r
summary(chair_clean$Sales)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   5.545   7.525   7.560   9.322  16.270
```

```r
#data set sucio, lleno de errores que trataremos, al que llamamos chairs_raw
chairs_raw <- read.csv(file="ChildCarSeats_brut.csv", sep = ",", dec = ".", stringsAsFactors = FALSE)
#indicamos que tenga el punto como separador decimal,
#y que los textos los ponga como string y no como factor

head(chairs_raw,20) #visualizamos el resultado
```

```
##    Sales CompPrice   Income Advertising Population    Price ShelveLoc Age
## 1   9500      138$  73,500k       11000        276     120$         1  42
## 2  11220      111$       48       16000      260 k      83$         3  65
## 3  10060      113$       35       10250      269 k      80$         2  59
## 4   7400      117$      100        5125        466      97$         2  55
## 5   4150 119.85EUR       64        3750        340 108.8EUR         1  38
## 6  10810      124$ 113,125k       13500        501      72$         1  78
## 7   6630  97.75EUR 105,125k           0         45  91.8EUR         2  71
## 8  11850      136$  81,500k       15000        425     120$         3  67
## 9   6540  112.2EUR      110         250        108 105.4EUR         2  76
## 10  4690      132$ 113,500k           0        131     124$         2  76
## 11 9.01K      121$  78,750k        9500        150     100$         1  26
## 12 11960      117$  94,750k        4000        503      94$         3  50
## 13  3980  103.7EUR       35        2125        393 115.6EUR         2  62
## 14 10960      115$       28       11000         29      86$         3  53
## 15  <NA>      107$ 117,750k       11375        148     118$         3  52
## 16 8.71K 126.65EUR  95,500k        5000        400 122.4EUR         2  76
## 17  7580  100.3EUR       32         125        284  93.5EUR         3  63
## 18 12290      147$  74,500k       13000      251 k     131$         3  52
## 19 13910      110$      110           0      408 k      68$         3  46
## 20  8730      129$  76,125k       16000       58 k     121$         2  69
##    Education Urban     US
## 1         17 urban     US
## 2         10 urban     US
## 3         12 urban    EUA
## 4         14 urban     US
## 5         13 urban non US
## 6         16 rural     US
## 7         15 urban non US
## 8         10 urban     US
## 9         10 rural non US
## 10        17 rural     US
## 11        10 rural     US
## 12        13 urban     US
## 13        18 urban non US
## 14        18 urban     US
```

```
## 15         18   Urb     US
## 16         18 rural non US
## 17         13   Urb non US
## 18         10 urban     US
## 19         17 rural     US
## 20         12 urban     US
```

2 Integracion y seleccion de los datos

realizamos la lectura del archivo a tratar, donde vemos que hay muchos errores lo leemos con el metodo read.csv ya que es un archivo csv delimitado por comas Sales y population deberia ser integer, compPrice deberia ser numeric, income, price, advertising tambien, shelveLoc deberia ser factor. Tambien vemos que algunos tienen el simbolo de euro y otros del dollar, tambien vemos que unos tienen un decimal con coma y otros con puntos, vemos que hay el simbolo k para simbolizar mil, etc, esto puede ser debido a tener diversas fuentes de datos, puede deberse a diferentes sistemas de notacion.

3 Limpieza de los datos

3.0 transformacion de clases y limpieza de elementos

Para empezar transformaremos los euros a dollares y eliminaremos el simbolo de las varaibles comprice y de Price

```
#cambiaremos los euros a dollares
for (i in seq_along(chairs_raw$CompPrice)) {
#relizamos un for que recorra todos los elementos de comprice
  if (str_detect(chairs_raw$CompPrice[i], "EUR")){
#dentro poner un if donde detecte para cada elemento si tiene la palabra eur

    chairs_raw$CompPrice[i] <- gsub("EUR", "", chairs_raw$CompPrice[i])
#en el caso que la tenga, quitara la palabra eur solo de ese numero

    euros <- as.numeric(chairs_raw$CompPrice[i])
#convertiremos a numerico guardado en otra variable llamada euro,
#solamente ese numero
    euros <- euros*0.82 #realizaremos la operacion de conviertirlo en dolares
    euros <- as.character(euros) #y volveremos a cambiar a caracter
    chairs_raw$CompPrice[i] <- euros}} #introduccimos el resultado

chairs_raw$CompPrice<- gsub('\\$', "", chairs_raw$CompPrice)
#como ya no hay euros, solamente quitamos los $
chairs_raw$CompPrice <- as.integer(chairs_raw$CompPrice)
#convertimos todo a integer, asi ya quedan redondeados los decimales
head(chairs_raw$CompPrice,20)
```

```
##  [1] 138 111 113 117  98 124  80 136  92 132 121 117  85 115 107 103  82
## [18] 147 110 129
```

```
#visualizamos para ver que efectivamente se ha realizado el cambio
```

Realizamos lo mismo para PRICE

```
for (i in seq_along(chairs_raw$Price)) {
  if (str_detect(chairs_raw$Price[i], "EUR")){
    chairs_raw$Price[i] <- gsub("EUR", "", chairs_raw$Price[i])

    euros <- as.numeric(chairs_raw$Price[i])
    euros <- euros*0.82
    euros <- as.character(euros)
    chairs_raw$Price[i] <- euros}}

chairs_raw$Price<- gsub('\\$', "", chairs_raw$Price)
chairs_raw$Price<- as.integer(chairs_raw$Price)
head(chairs_raw$Price,20)
```

```
##  [1] 120  83  80  97  89  72  75 120  86 124 100  94  94  86 118 100  76
## [18] 131  68 121
```

```
#vemos que ha convertido todo a dollares quitando el simbolo
```

Ahora tratamos la variable SALES donde vamos a eliminar la letra k,

```
chairs_raw$Sales <- chairs_raw$Sales <- gsub("k", "", chairs_raw$Sales)
#usamos la funcion gsub para sustituir el caracter k por nada,
chairs_raw$Sales <- as.numeric(chairs_raw$Sales)
```

```
## Warning: NAs introducidos por coerción
```

```
#pasamos a numeric la columna sales, que ahora mismo es characters
#tenemos que pasar los NA a 0 ya que sino al recorrer la variable nos da error
sum(is.na(chairs_raw))
```

```
## [1] 51
```

```
#vemos que existes 51 NA, y que esta ubicados todos en Sales
sapply(chairs_raw, function(x) sum(is.na(x)))
```

```
##       Sales   CompPrice      Income Advertising  Population       Price
##          51           0           0           0           0           0
##   ShelveLoc         Age   Education       Urban          US
##           0           0           0           0           0
```

```
chairs_raw$Sales[is.na(chairs_raw$Sales)] <- 0.01 #cambiamos los NA a ceros

for (i in seq_along(chairs_raw$Income)) {
#realizamos un for que recorra todos los elementos y
#con el if esocgemos los que son mayores que 1000 a estos los
#dividimos por 1000 para igualar las notaciones
  if (chairs_raw$Sales[i]>1000) {
    chairs_raw$Sales[[i]] <- chairs_raw$Sales[[i]]/1000}}
chairs_raw$Sales <- round(chairs_raw$Sales, 2)
#mediante la funcion round indicamos que la redondee a 2 decimales
head(chairs_raw$Sales,20) #mostramos el resultado
```

```
##  [1]  9.50 11.22 10.06  7.40  4.15 10.81  6.63 11.85  6.54  4.69  0.01
## [12] 11.96  3.98 10.96  0.01  0.01  7.58 12.29 13.91  8.73
```

tratamos ahora la variable INCOME

```
chairs_raw$Income <- gsub("k", "", chairs_raw$Income)
#eliminamos las k de la columna Income
chairs_raw$Income <- gsub("\\.", "", chairs_raw$Income)
#tambien eliminamos el punto que le ponemsos \\
#ya que sino indicaria todos los elementos
chairs_raw$Income <- gsub(",", "", chairs_raw$Income)
#eliminamos las comas

chairs_raw$Income <- as.numeric(chairs_raw$Income)
#convertimos los caracteres como numeros para poder operar con ellos
for (i in seq_along(chairs_raw$Income)) {
  #realizamos un for que recorra todos los elementos y con el
  #if escogemos los que son mayores que 1000 a estos los dividimos
  #por 1000 para igualar las notaciones
  if (chairs_raw$Income[i]>1000) {
    chairs_raw$Income[[i]] <- chairs_raw$Income[[i]]/1000}}
chairs_raw$Income <- trunc(chairs_raw$Income)
#truncamos el resultado dejando en enteros si hay algun decimal
head(chairs_raw$Income,20)
```

```
##  [1]  73  48  35 100  64 113 105  81 110 113  78  94  35  28 117  95  32
## [18]  74 110  76
```

```
#mostramos el resultado
```

tratamos POPULATION

```
chairs_raw$Population <- gsub("k", "", chairs_raw$Population)
#eliminamos las k
chairs_raw$Population <- gsub(" ", "", chairs_raw$Population)
#eliminamos los espacios en blanco que hay
chairs_raw$Population <- as.numeric(chairs_raw$Population)
#transformamos a numeric
chairs_raw$Population <- trunc(chairs_raw$Population)
#truncamos por si hay decimales
head(chairs_raw$Population,20) #vemos el resultado
```

```
##  [1] 276 260 269 466 340 501  45 425 108 131 150 503 393  29 148 400 284
## [18] 251 408  58
```

tratamos ADVERTISING

```
chairs_raw$Advertising <- as.numeric(chairs_raw$Advertising)
#pasamos a numeric la variable
chairs_raw$Advertising <- chairs_raw$Advertising/1000
#dividimos entre mil para que sean miles
chairs_raw$Advertising <- trunc (chairs_raw$Advertising)
```

```
#truncamos para eliminar los decimales
chairs_raw$Advertising <- as.integer(chairs_raw$Advertising)
#ahora que esta tratada la convertimos a integer
head(chairs_raw$Advertising,20) #vemos los primeros 20 resultados
```

```
##  [1] 11 16 10  5  3 13  0 15  0  0  9  4  2 11 11  5  0 13  0 16
```

```
class(chairs_raw$Advertising) #vemos la clase
```

```
## [1] "integer"
```

vemos que AGE y EDUCATION esta correcta por lo que no tratamos la variable

```
head(chairs_raw$Age,20)
```

```
##  [1] 42 65 59 55 38 78 71 67 76 76 26 50 62 53 52 76 63 52 46 69
```

```
#vemos que todo son enteros sin digitos decimales,
#asi que no realizamos ninguna accion


class(chairs_raw$Education)
```

```
## [1] "integer"
```

```
#vemos que es entera, por lo que no requiere más accion
```

ShelveLoc donde cambiamos a factor

```
chairs_raw$ShelveLoc <-as.factor(chairs_raw$ShelveLoc)
#convertimos a factor ShelveLoc
class(chairs_raw$ShelveLoc) #vemos que efectivamente es factor
```

```
## [1] "factor"
```

```
chairs_raw$ShelveLoc <- revalue(chairs_raw$ShelveLoc, c(
  "1"="Bad", "2"="Medium", "3"="Good"))
#reasignamos los valores 1 a bad, 2 a medium y 3 a good
levels(chairs_raw$ShelveLoc)
```

```
## [1] "Bad"    "Medium" "Good"
```

```
#vemos que efectivamente se ha realizado el cambio
```

Urban, vemos que en la variable urban hay disparidad de notacion para la misma informacion , urban escrito en mayusculas o escrito incompleto etc, lo mismo con city

```
chairs_raw$Urban <- gsub(" ", "", chairs_raw$Urban)
#eliminanos los espacios en blanco
chairs_raw$Urban <- gsub("\\bR\\w+", "rural", chairs_raw$Urban)
#todas las palabras que empiezan por R se susituyen completamente por rural
chairs_raw$Urban <- gsub("\\bU\\w+", "urban", chairs_raw$Urban)
#todas las que empiezas por U mayuscula por urban
chairs_raw$Urban <- gsub("\\bu\\w+", "urban", chairs_raw$Urban)
#todas las que empiezan por u por urban, aqui completamos todos los urb etc
chairs_raw$Urban <- gsub("city", "urban", chairs_raw$Urban)
#todas las coincidencias city por urban

chairs_raw$Urban <- as.factor(chairs_raw$Urban) #lo transformamos a factor
chairs_raw$Urban <- revalue(chairs_raw$Urban, c("urban" ="Yes", "rural"= "No"))
#cambiamos las etiquetas
levels(chairs_raw$Urban)
```

```
## [1] "No"  "Yes"
```

```
 #vemos que solo hay dos niveles
```

US vemos que tambien tiene disparidad de notacion

```
chairs_raw$US <- gsub(" ", "", chairs_raw$US)
#eliminanos los espacios en blanco
chairs_raw$US <- gsub("EUA", "US", chairs_raw$US)
#todas las palabras que tiene EUA y se sustituyen por US
chairs_raw$US <- gsub("USA", "US", chairs_raw$US)
#todas las palabras que tiene USA y se sustituyen por US

chairs_raw$US <- as.factor(chairs_raw$US)
#lo transformamos a factor
chairs_raw$US <- revalue(chairs_raw$US, c("US" ="Yes", "nonUS"= "No"))
#cambiamos las etiquetas
levels(chairs_raw$US)
```

```
## [1] "No"  "Yes"
```

```
#vemos que solo hay dos niveles
```

3.1 Valores nulos

Encontramos que poniendo si hay algun NA, nos dice si se encuentran, y con la funcion sum nos dice cuantos hay en total

```
any(is.na(chairs_raw)) #nos dice si hay NA
```

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

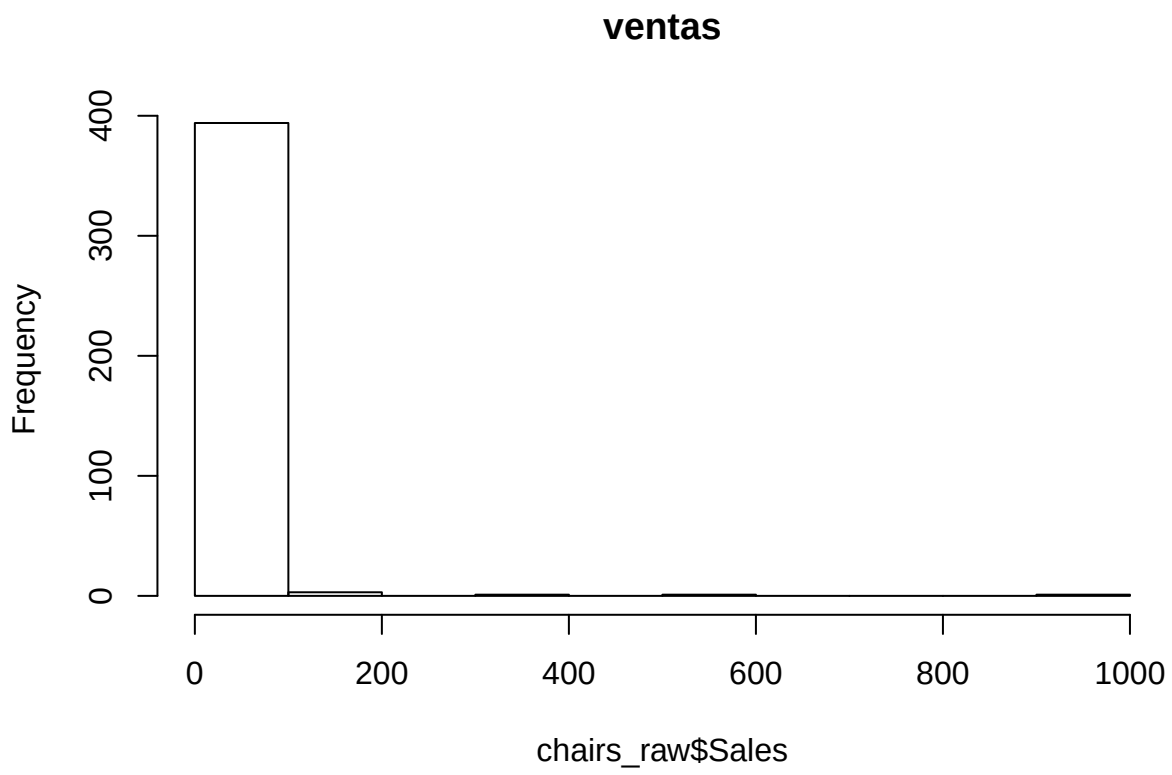
```r
sum(is.na(chairs_raw)) #nos dice cuantos hay, no deberia haber ya que hemos cambiado los NA por ceros,
```

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

3.2 valores extremos

se define como valor atipico leve aquel que dista 1,5 veces el rango intercuartilico por debajo de Q1 o por encima de Q3 y valor atipico extremo el que dista 3 veces por lo que q3 (9.332) + 3 *IQR (3.774)= 20,642, eliminaremos por tanto todos los valores superiores a este, y por abajo sera q1 (5.558) -3* IQR (3.774)= -5,702 es el valor minimo, como es negativo y el minimo que tenemos es 0, no superaremos el umbral aun asi pondremos NA a los 0 ya que los transformamos al principio del ejercicio para la variable sales

```r
hist(chairs_raw$Sales, main = "ventas")
```




```r
#creamos un histograma que nos muestre la frecuencia de ventas
#vemos que la gran mayoria estan situados de 0 a 200
chairs_raw$Sales[chairs_raw$Sales == 0.01] <- NA
#volvemos a poner los 0.01 por NA, ya que si no nos afectara a nuestros calculos
summary(chairs_raw$Sales)
```

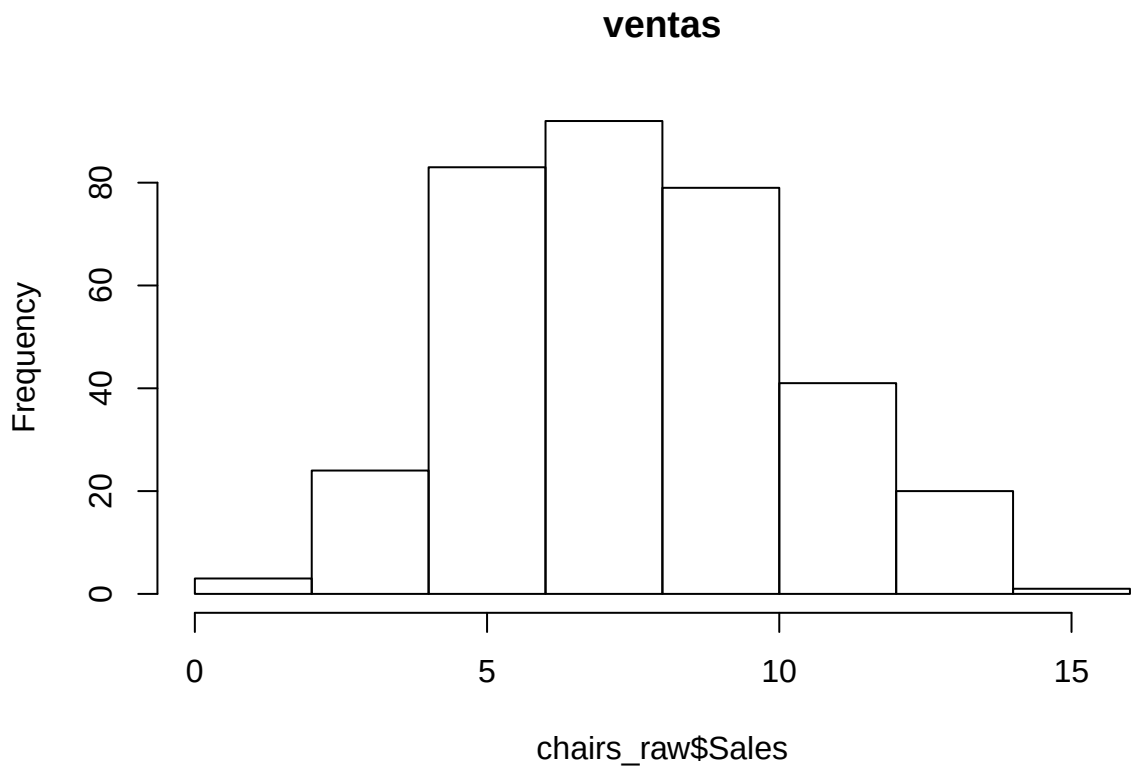
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    5.55    7.50   13.70    9.33  910.00      51
```

```r
#nos muestra y agrupa por cuartos y su media
# vemos que si 1st quarter esta en 5.558 y que el 3rd quarter esta en 9.332
#IQR = Q3 -Q1 = 9.332 - 5.558 = 3.774


chairs_raw$Sales[chairs_raw$Sales > 20] <- NA
#cambiamos por NA los numeros mayores a 20
head(chairs_raw$Sales) #vemos el resultado
```

```
## [1]  9.50 11.22 10.06  7.40  4.15 10.81
```

```r
hist(chairs_raw$Sales, main = "ventas")
```




```r
#volvemos a realizar un histograma pero ahora con los datos corregidos y
#vemos que obtenemos una campana de gauss
summary(chairs_raw$Sales)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   0.000   5.525   7.450   7.499   9.275  15.630      57
```

Consideramos que el archivo esta ya limpio

```
chairs_net <- chairs_raw
head(chairs_net, 20)
```

```
##    Sales CompPrice Income Advertising Population Price ShelveLoc Age
## 1   9.50       138     73          11        276   120       Bad  42
## 2  11.22       111     48          16        260    83      Good  65
## 3  10.06       113     35          10        269    80    Medium  59
## 4   7.40       117    100           5        466    97    Medium  55
## 5   4.15        98     64           3        340    89       Bad  38
## 6  10.81       124    113          13        501    72       Bad  78
## 7   6.63        80    105           0         45    75    Medium  71
## 8  11.85       136     81          15        425   120      Good  67
## 9   6.54        92    110           0        108    86    Medium  76
## 10  4.69       132    113           0        131   124    Medium  76
## 11    NA       121     78           9        150   100       Bad  26
## 12 11.96       117     94           4        503    94      Good  50
## 13  3.98        85     35           2        393    94    Medium  62
## 14 10.96       115     28          11         29    86      Good  53
## 15    NA       107    117          11        148   118      Good  52
## 16    NA       103     95           5        400   100    Medium  76
## 17  7.58        82     32           0        284    76      Good  63
## 18 12.29       147     74          13        251   131      Good  52
## 19 13.91       110    110           0        408    68      Good  46
## 20  8.73       129     76          16         58   121    Medium  69
##    Education Urban  US
## 1         17   Yes Yes
## 2         10   Yes Yes
## 3         12   Yes Yes
## 4         14   Yes Yes
## 5         13   Yes  No
## 6         16    No Yes
## 7         15   Yes  No
## 8         10   Yes Yes
## 9         10    No  No
## 10        17    No Yes
## 11        10    No Yes
## 12        13   Yes Yes
## 13        18   Yes  No
## 14        18   Yes Yes
## 15        18   Yes Yes
## 16        18    No  No
## 17        13   Yes  No
## 18        10   Yes Yes
## 19        17    No Yes
## 20        12   Yes Yes
```

Ahora realizamos la sustitucion de valores NA por la media Para ello usamos el chairs_raw poder tratarla sin afectar a la muestra

4 Analisis de los datos

Despues de estudiar los objetivos de las muestras, miraremos si la variable Sales es de distribución normal, para ello realizaremos un contraste de hipotesis donde crearemos una hipotesis nula y una alternativa donde:

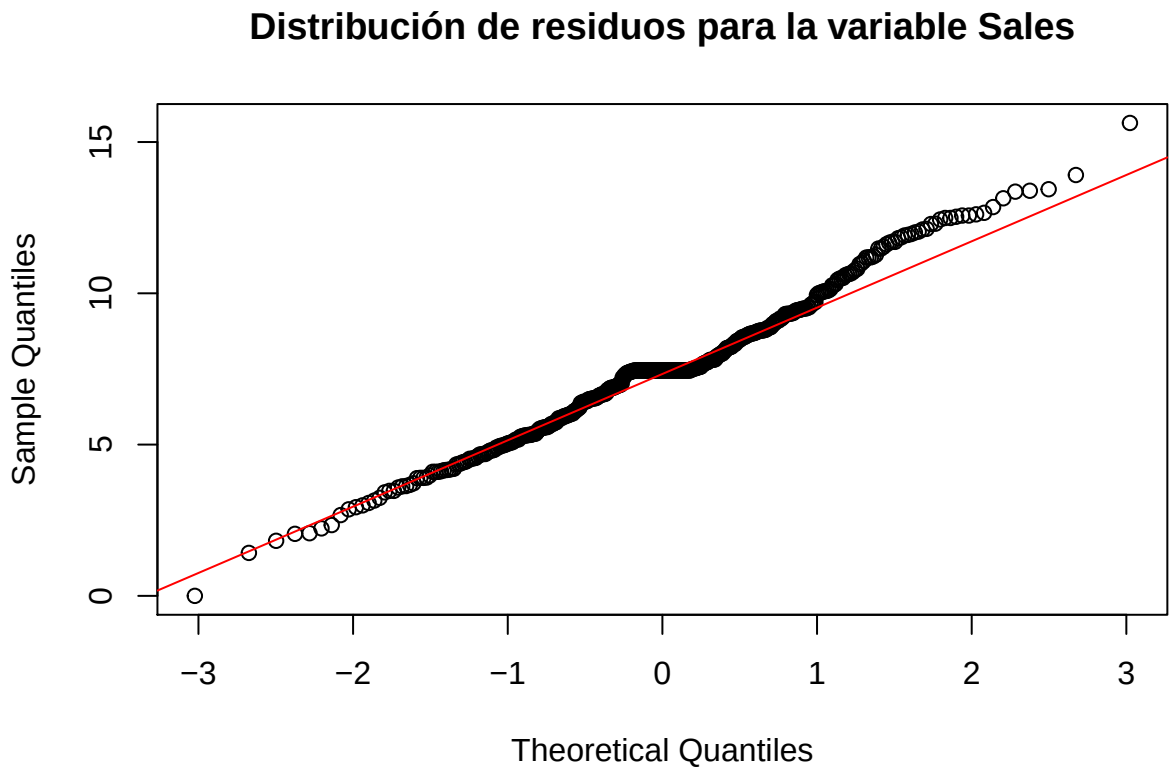
H0 : La muestra proviene de una distribución normal H1 : La muestra no proviene de una distribución normal

El nivel de confianza sera siempre del 95 por ciento por lo que alpha sera 0.05 y donde si P < Alpha entonces se rechaza H0 si p >= Alpha entonces no se rechaza H0

```
chairs_raw$Sales<-na.replace(chairs_raw$Sales,median(chairs_raw$Sales, na.rm = TRUE))

#remplazamos los NA por  la media de toda la variable Sales

c<-qqnorm(chairs_raw$Sales,
        main = "Distribución de residuos para la variable Sales")
qqline(chairs_raw$Sales, col = 2)
```




```
#realizamos un grafico q-q de residuos para ver la normalidad, vemos
#que tenemos un quiebro en medio
ad.test(chairs_raw$Sales)
```

```
##
##  Anderson-Darling normality test
##
## data:  chairs_raw$Sales
## A = 2.329, p-value = 6.613e-06
```

```r
shapiro.test(chairs_raw$Sales)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  chairs_raw$Sales
## W = 0.98762, p-value = 0.001767
```

```r
lillie.test(chairs_raw$Sales)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  chairs_raw$Sales
## D = 0.08568, p-value = 1.93e-07
```

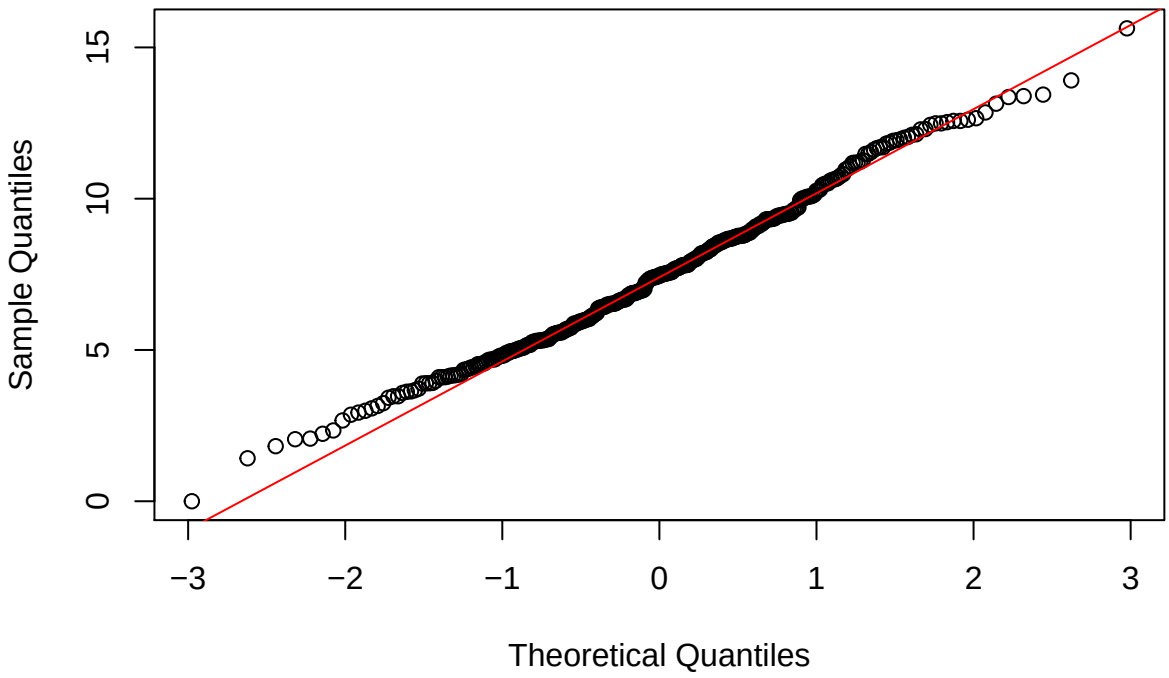
```r
#realizamos 3 diferentes test para comprobar la normalidad
#y vemos que no cumple la normalidad ya que p value es muy pequeño
```

Probamos ahora sin sustituir los NA por la media, ya que puede distorsionar los resultados Vemos que en este caso la muestra se puede considerar normal ya que en 2 de las 3 pruebas esta por encima de 0.05 y en ad.test esta casi a 0.05

```r
delete.na <- function(df, n=0) {
 df[rowSums(is.na(df)) <= n,]
}
chairs_net <- delete.na(chairs_net)
#eliminamos los NA de la muestra

write.csv(chairs_net, file="chairs_net.csv")
#guardamos el archivo limpio como csv
c<-qqnorm(chairs_net$Sales,
        main = "Distribución de residuos para la variable Sales")


qqline(chairs_net$Sales, col = 2)
```




```
#realizamos un grafico q-q de residuos para ver la normalidad, vemos
#que ya no tenemos un quiebro en medio
ad.test(chairs_net$Sales)
```

```
##
##  Anderson-Darling normality test
##
## data:  chairs_net$Sales
## A = 0.77595, p-value = 0.04351
```

```
shapiro.test(chairs_net$Sales)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  chairs_net$Sales
## W = 0.99282, p-value = 0.09938
```

```
lillie.test(chairs_net$Sales)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  chairs_net$Sales
## D = 0.039961, p-value = 0.2019
```

```r
#realizamos 3 diferentes test para comprobar la normalidad
#y vemos que cumple la normalidad ya que p value es mas grande de 0.05 en 2 de
#los tres metodos
```

Intervalo de confianza calculamos la funcion confidence

```r
#creamos la funcion confidence que nos dara el intervalo de confianza de la
#media poblacional de la variable sales
#consideramos que el nivel de confianza sera del 95 por ciento
funcion_confidence = function(x) {
  right = mean(x) + qnorm(.975)*(sd(x))/(sqrt(length(x))) #calculamos los dos
  #lados, y sera la media mas el error
  left = mean(x) - qnorm(.975)*(sd(x))/(sqrt(length(x))) #en este caso sera
  #la media mas el error
  print(right) #imprimimos los resultados de ambos lados
  print(left)

}
funcion_confidence(chairs_net$Sales) #llamamos a la funcion y la comparamos
```

```
## [1] 7.779142
## [1] 7.218817
```

```r
#con la funcion t.test, y vemos que son iguales
t.test(chairs_net$Sales)
```

```
##
##  One Sample t-test
##
## data:  chairs_net$Sales
## t = 52.461, df = 342, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  7.217822 7.780137
## sample estimates:
## mean of x
##   7.49898
```

intervalo de confianza Sales Us y Sales no US

```r
# reutilizamos la funcion_confidence que hemos realizado anteriormente,
# creamos dos subgurpos los que se han vendido en US y los que no

confi_US= chairs_net[chairs_net$US == "Yes",]
#llamamos a la funcion con los parametros de ventas de US
funcion_confidence(confi_US$Sales)
```

```
## [1] 8.372543
## [1] 7.63131
```

```r
confi_noUS = chairs_net[chairs_net$US == "No",]
#llamamos a la funcion con los parametros de ventas de US
funcion_confidence(confi_noUS$Sales)
```

```
## [1] 6.992403
## [1] 6.251277
```

```r
#creo que las medias poblacionales de las dos muestras son diferentes,
#ya que hay una diferencia del 20 por ciento aproximadamente
```

Estudiamos ahora las ventas en US y fuera de US Hipotesis nula y alternativa Realizamos la hipotesis nula Donde H0 : m1 = m2 donde m1 son las ventas en las tiendas de US y m2 las ventas fuera de US y la hipotesis alternativa H1 : m1 > m2

Calculos Puesto que hemos considerado que la muestra tiene una distribucion normal,

```r
#determinamos el nivel de significación
alpha = 0.05
#calculamos la desviacion estandar de las dos muestras
sd_US = sd(confi_US$Sales)
sd_noUS = sd(confi_noUS$Sales)
var_US = var(confi_US$Sales)
var_noUS = var(confi_noUS$Sales)
#calculamos el tamaño de las muestras
n_US = nrow(confi_US)
n_noUS = nrow(confi_noUS)
#grados de libertad
v_US = n_US -1
v_noUS = n_noUS -1
#calculamos la media de las muestras
mean_US = mean(confi_US$Sales)
mean_noUS = mean(confi_noUS$Sales)
#sumas de cuadrados de diferencias
ss_US = sum((confi_US$Sales - mean(confi_US$Sales)^2))
ss_noUS =  sum((confi_noUS$Sales - mean(confi_noUS$Sales)^2))
#varianza agrupada
s2p = (ss_US +ss_noUS)/(v_US + v_noUS)
#error estandar de la diferencia de medias
e_standard = sqrt((var_US/n_US)+(var_noUS/n_noUS))
#calculamos el estadistico de contraste


z = (mean_US - mean_noUS) / e_standard
z
```

```
## [1] 5.161145
```

Conclusiones

```r
#calculamos el p valor usando la funcion pnorm con el estadistico de contraste
pValor = 1 - pnorm(z)
pValor
```

```
## [1] 1.227218e-07
```

```
#comprobamos ahora que usando la funcion test, obtenemos el mismo t, y que
# obtenemos un resultado del p valor del mismo orden de magnitud, donde podemos
#ver que es menor que 0.05, por lo que rechazamos la hipotesis nula
#y llegamos a la conclusion que las ventas en US son mayores que las ventas
#fuera
t.test(confi_US$Sales, confi_noUS$Sales,
       alternative = "greater", conf.level = 0.95)
```

```
##
##  Welch Two Sample t-test
##
## data:  confi_US$Sales and confi_noUS$Sales
## t = 5.1611, df = 315.66, p-value = 2.175e-07
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.9389593       Inf
## sample estimates:
## mean of x mean of y
##  8.001927  6.621840
```

```
#por lo que podemos decir que las medias poblacionales calculadas
# son diferentes
```

Ventas en zonas urbanas y rurales Hipotesis Realizamos la hipotesis nula Donde H0 : m1 = m2 donde m1 son las ventas en zonas urbanas y m2 las ventas en zonas rurales y la hipotesis alternativa H1 : m1 != m2

```
# reutilizamos la funcion_confidence que hemos realizado anteriormente,
# creamos dos subgurpos los que se han vendido en zona urbana  y los que no

confi_urban= chairs_net[chairs_net$Urban == "Yes",]
#llamamos a la funcion con los parametros de ventas de US
funcion_confidence(confi_urban$Sales)
```

```
## [1] 7.746364
## [1] 7.078136
```

```
confi_rural = chairs_net[chairs_net$Urban == "No",]
#llamamos a la funcion con los parametros de ventas de US
funcion_confidence(confi_rural$Sales)
```

```
## [1] 8.215577
## [1] 7.186558
```

```
#creo que las medias poblacionales de las dos muestras son iguales ya que no
#hay casi diferencia y la izquierda de las dos (urbana y rural) son
#practicamente identicas,
```

Puesto que hemos considerado que la muestra tiene una distribucion normal,

```r
#determinamos el nivel de significación
alpha = 0.05
#calculamos la desviacion estandar de las dos muestras
sd_urban = sd(confi_urban$Sales)
sd_rural = sd(confi_rural$Sales)
var_urban = var(confi_urban$Sales)
var_rural = var(confi_rural$Sales)
#calculamos el tamaño de las muestras
n_urban = nrow(confi_urban)
n_rural = nrow(confi_rural)
#grados de libertad
v_urban = n_urban -1
v_rural = n_rural -1
#calculamos la media de las muestras
mean_urban = mean(confi_urban$Sales)
mean_rural = mean(confi_rural$Sales)

#error estandar de la diferencia de medias
e_standard = sqrt((var_urban/n_urban)+(var_rural/n_rural))
#calculamos el estadistico de contraste


z = (mean_urban - mean_rural) / e_standard
z
```

```
## [1] -0.9227306
```

```r
#calculamos el p valor usando la funcion pnorm con el estadistico de contraste
pValor = 1 - pnorm(z)
pValor
```

```
## [1] 0.8219262
```

```r
#otenemos que p value es de 0.822, donde podemos
#ver que es mayor que 0.05, por lo que no podemos rechazar la hipotesis
#nula, y debemos aceptar que las ventas en zonas urbanas no son diferentes de
#las zonas rurales
```

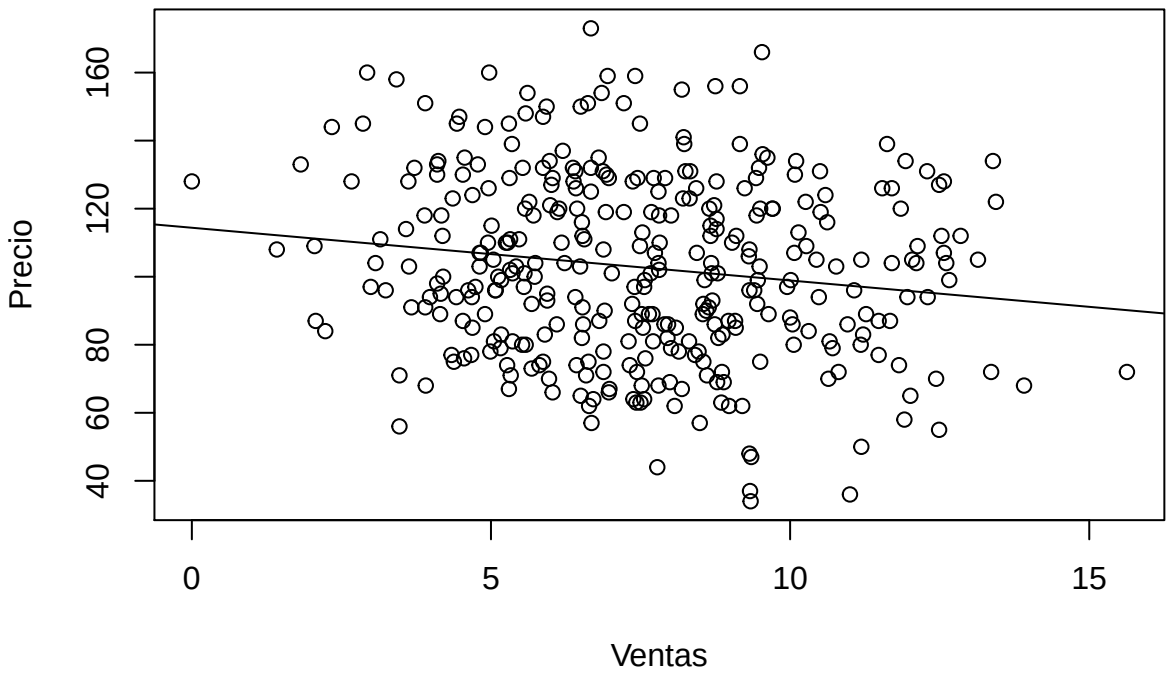
Realizaremos ahora estudios para ver la importancia de las variables mediante modelos de regresión lineal

```r
# realizamos la regresion lineal por minimos cuadrados de Sales
#en funcion de Price que significa como cambian las ventas de sillitas
#por la variacion del precio de venta
#mediante la funcion lm vemos la ecuacion de la recta
ventas_precio = lm(chairs_net$Price~chairs_net$Sales)
summary(ventas_precio)
```

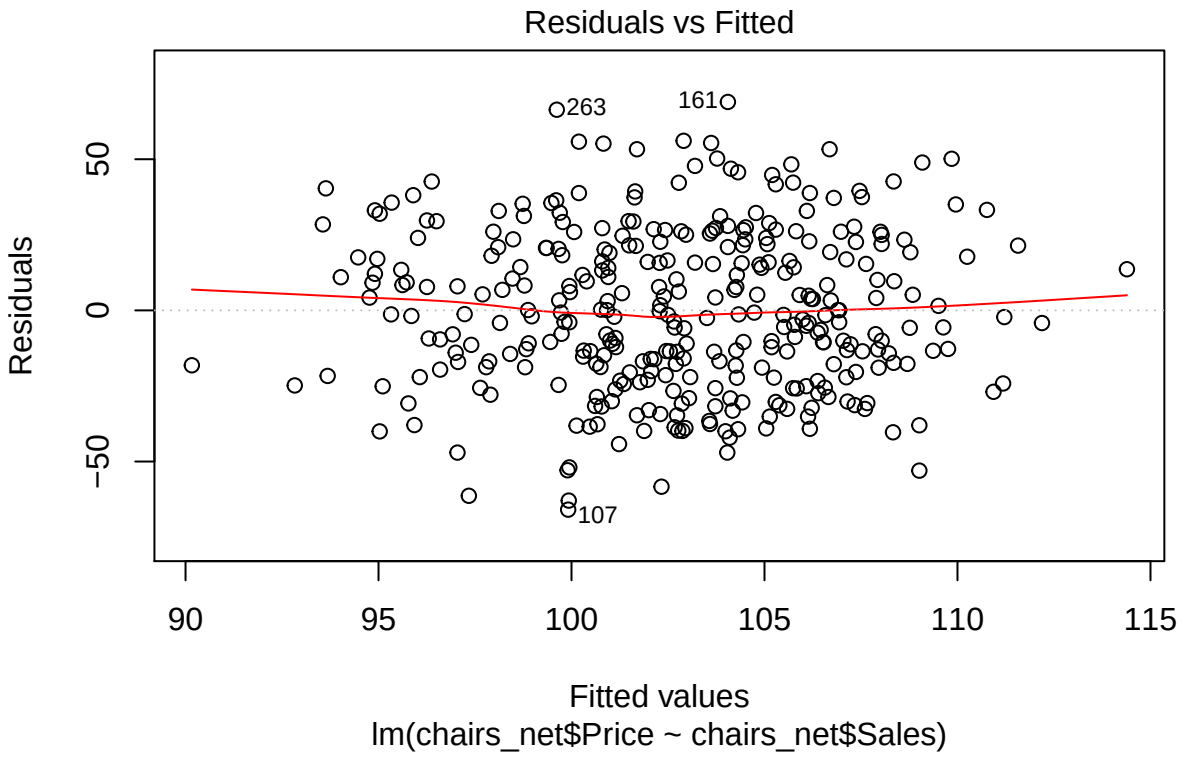
```
##
## Call:
## lm(formula = chairs_net$Price ~ chairs_net$Sales)
##
## Residuals:
```

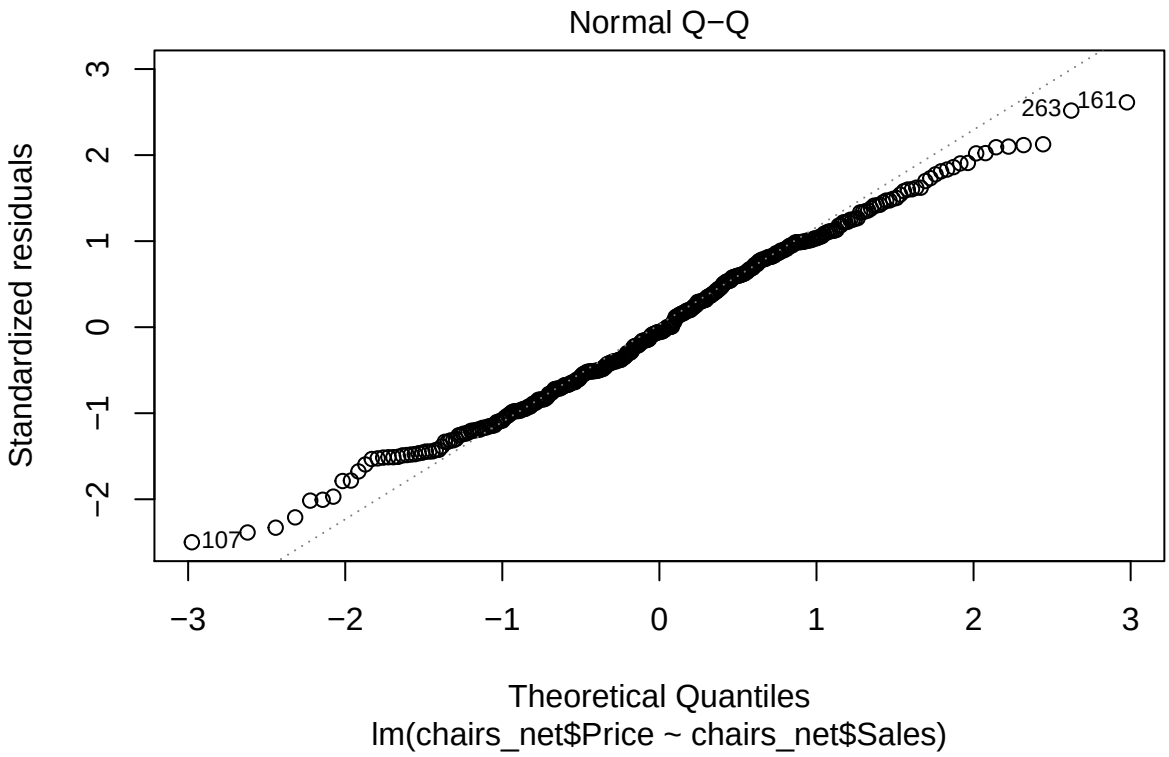
```
##     Min      1Q  Median      3Q     Max
## -65.913 -19.277  -1.493  20.924  68.948
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      114.3905     4.2922  26.651  < 2e-16 ***
## chairs_net$Sales  -1.5500     0.5398  -2.871  0.00434 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 26.43 on 341 degrees of freedom
## Multiple R-squared:  0.02361,    Adjusted R-squared:  0.02075
## F-statistic: 8.245 on 1 and 341 DF,  p-value: 0.004342
```

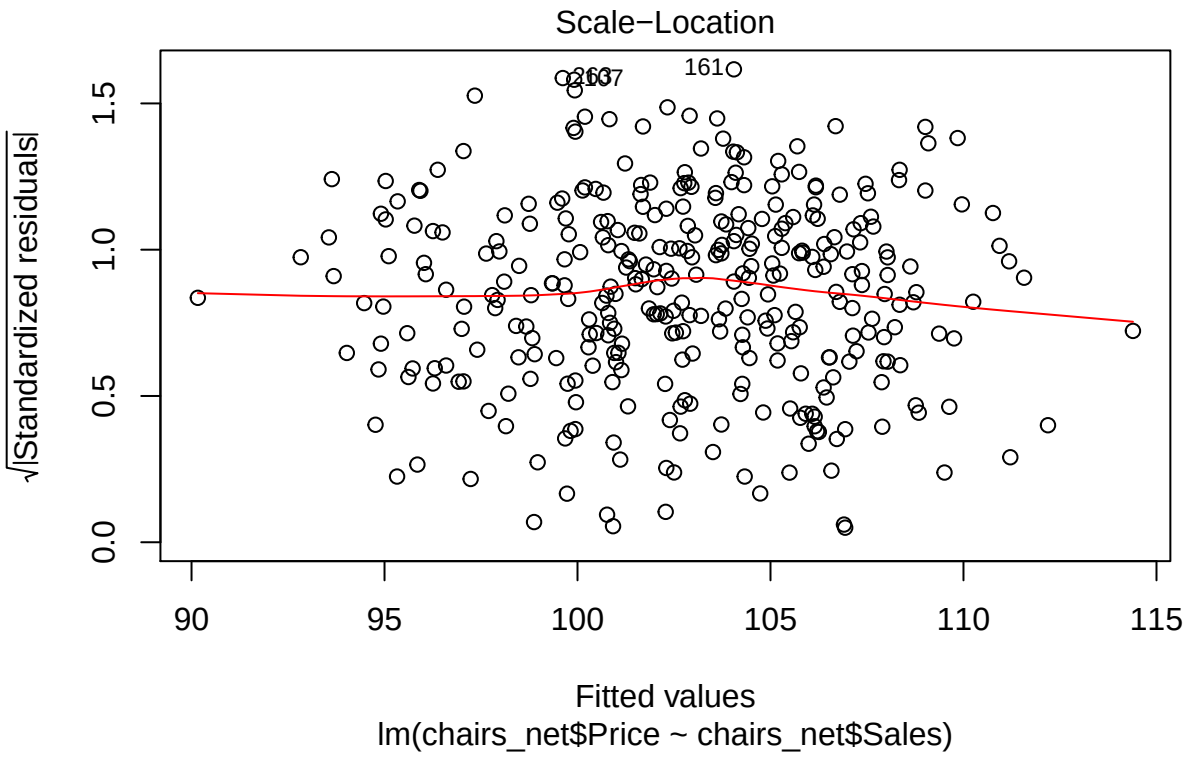
```r
#usamos la funcion summary para ver los resultados
#que aparecen en el estimate , el intercept es la coincidencia en el 0 de
#las dos variables lo que en una recta seria la ordenada en el origen ,
#y el estimate de la variable, indica
#la pendiente de la recta.
#que en este caso es
#- 1.55x + 114.3905 lo que indica que tiene pendiente negativa,
#a medida que el precio disminuye las ventas suben
#aunque es muy poco pronunciada
# la r^2 que representa la variabilidad observada en la variable,
#en este caso es muy baja 0.02
#lo que significa que no es una variable representativa para el
#comportamiento de los precios el p value es significativamente bajo,
#considerando que nuestro nivel de confianza,
#por lo tanto no aceptable para una p mayor que 0.05,
#en este caso la p es 0.004, por lo que debemos aceptar el resultado
plot(chairs_net$Sales, chairs_net$Price, xlab = "Ventas"
     , ylab="Precio")
#representamos un diagrama de dispresion
abline(ventas_precio)
```

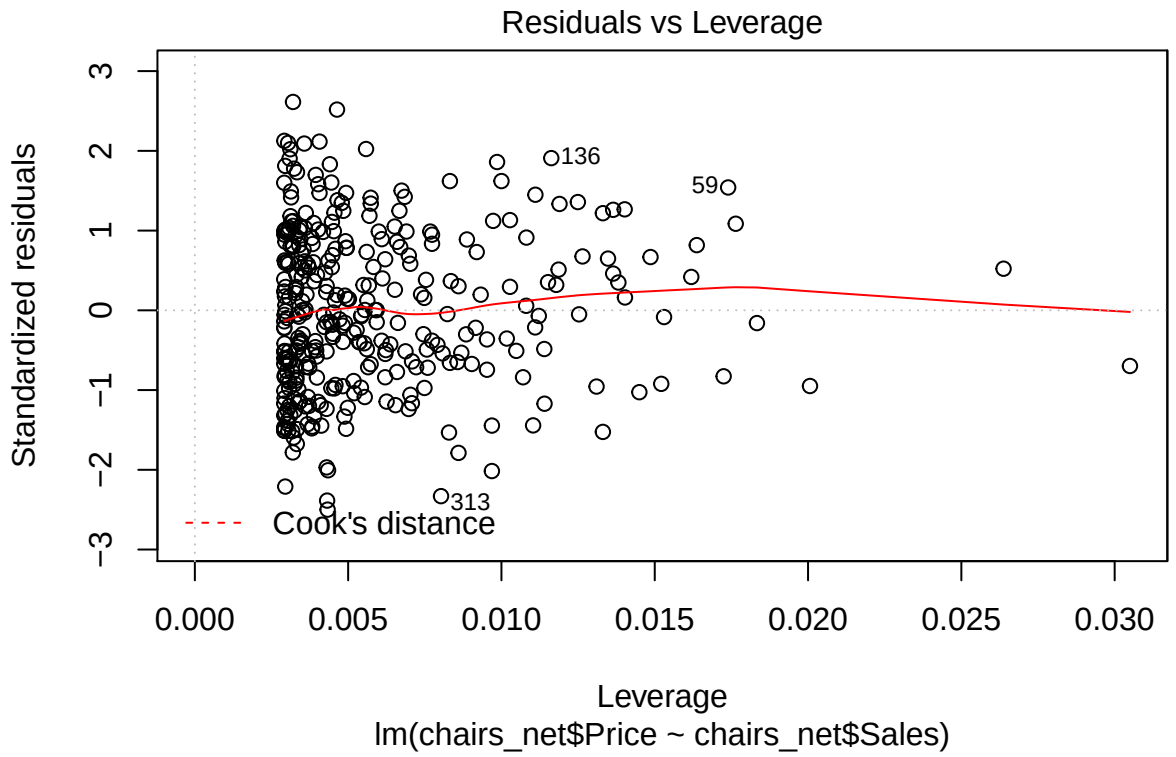



```
##vemos que la pendiente es negativa,
#aunque no se puede ver graficamente que el comportamiento
#sea estrictamente lineal
plot(ventas_precio)
```


Residuals vs Fitted
263
161
107
Residuals
50
0
−50
90
95
100
105
110
115
Fitted values
lm(chairs_net$Price ~ chairs_net$Sales)


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(chairs_net$Price ~ chairs_net$Sales)
107
263
161
−3
−2
−1
0
1
2
3


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
90
95
100
105
110
115
Fitted values
lm(chairs_net$Price ~ chairs_net$Sales)




```
#si realizamos el plot de la regresion lineal, vemos que
#en el grafico resudiual fitted
#no vemos ninguna tendencia lo que la homocedasticidad
#y la linealidad resultan aceptables
#en el caso del grafico normal q- q , podemos ver una tendencia lineal
#en cierta parte de las muestras
```

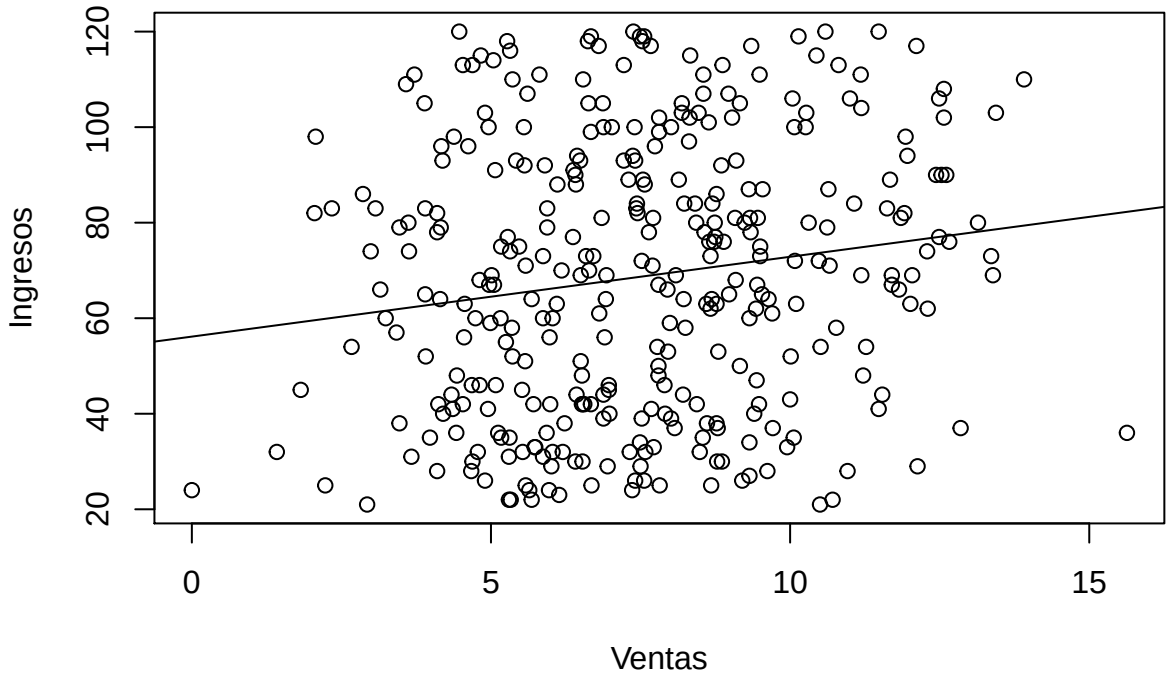
Realizamos ahora el mismo procedimiento donde estudiaremos las ventas en funcion de los ingresos de los compradores

```
ventas_Ingresos = lm(chairs_net$Income~chairs_net$Sales)
summary(ventas_Ingresos)
```

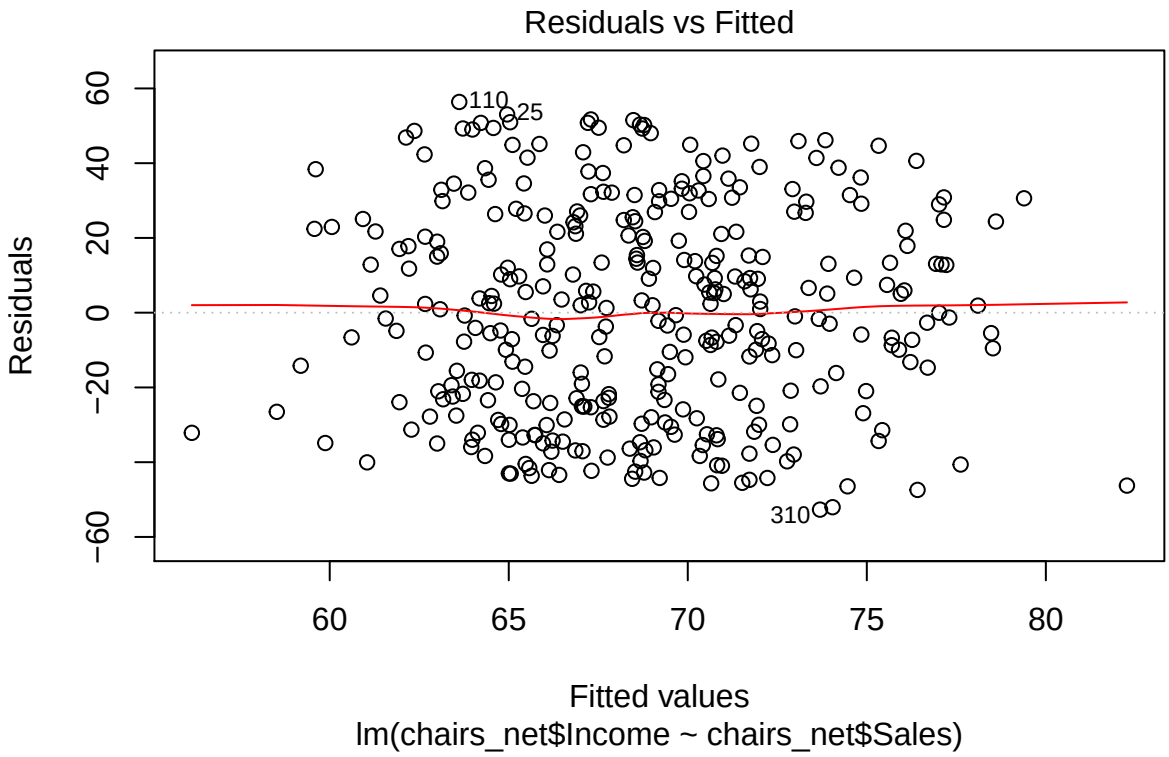
```
##
## Call:
## lm(formula = chairs_net$Income ~ chairs_net$Sales)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -52.694 -24.540  -0.667  23.041  56.382
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       56.1493     4.5447  12.355  < 2e-16 ***
## chairs_net$Sales   1.6709     0.5716   2.923  0.00369 **
```

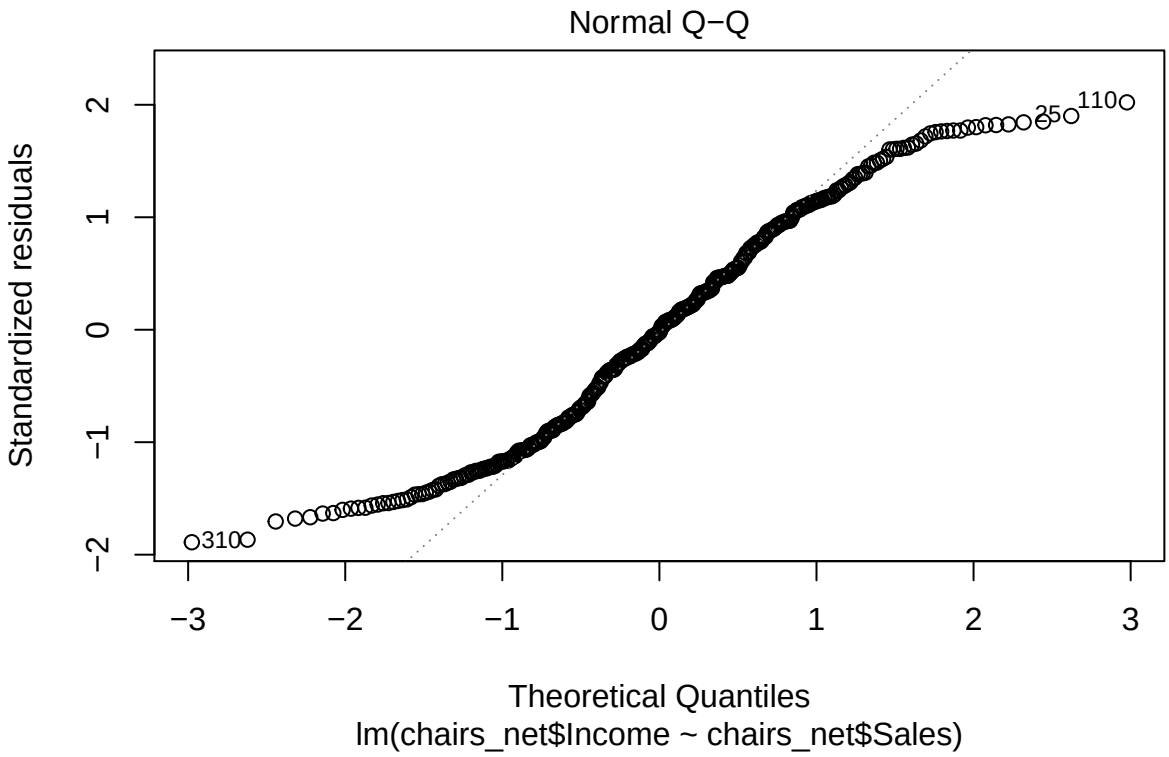
```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 27.98 on 341 degrees of freedom
## Multiple R-squared:  0.02445,    Adjusted R-squared:  0.02159
## F-statistic: 8.546 on 1 and 341 DF,  p-value: 0.003694
```

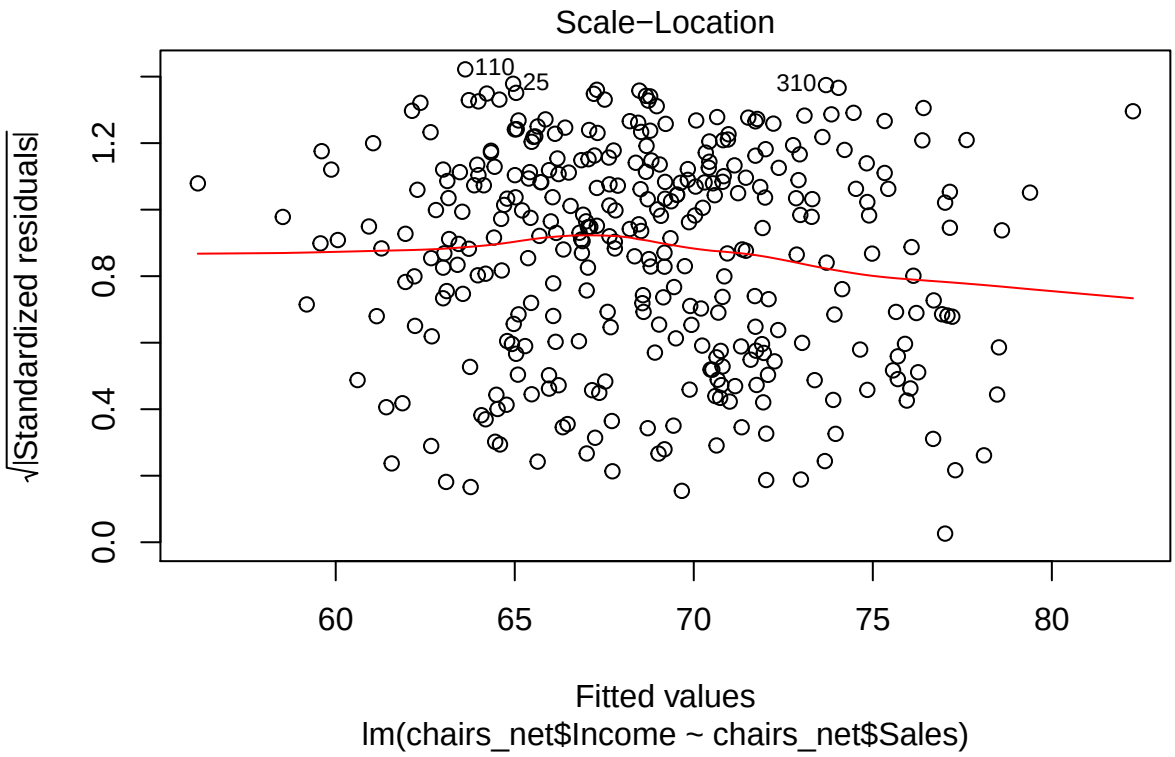
```r
plot(chairs_net$Sales, chairs_net$Income, xlab = "Ventas"
     , ylab="Ingresos")
#representamos un diagrama de dispresion
abline(ventas_Ingresos)
```

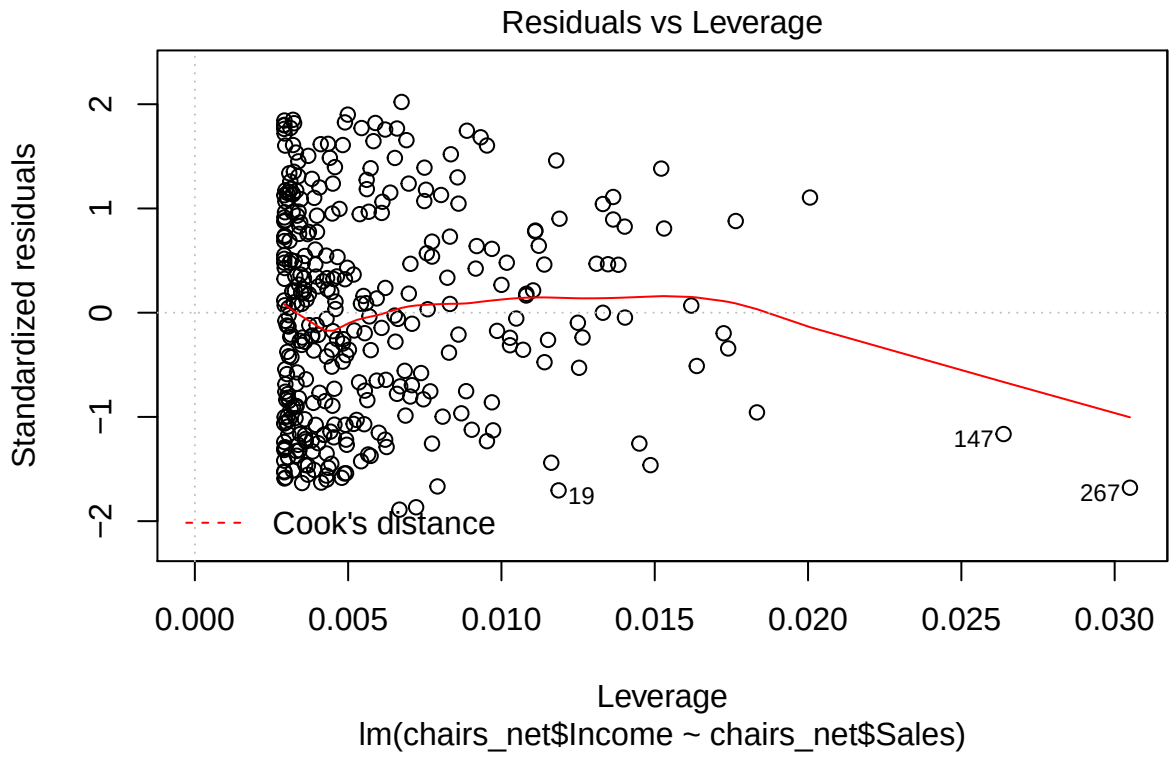



```r
##vemos que la pendiente es positiva,
#aunque no se puede ver graficamente que el comportamiento
#sea estrictamente lineal
plot(ventas_Ingresos)
```


Residuals vs Fitted
Residuals
Fitted values
lm(chairs_net$Income ~ chairs_net$Sales)
110
25
310
60
40
20
0
−20
−60
60
65
70
75
80


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(chairs_net$Income ~ chairs_net$Sales)
310
25
110
−3
−2
−1
0
1
2
3


Scale−Location
110
25
310
√|Standardized residuals|
0.0
0.4
0.8
1.2
60
65
70
75
80
Fitted values
lm(chairs_net$Income ~ chairs_net$Sales)



realizareos ahora un Modelo de regresion lineal multiple

```
#estimamos por minimos cuadrados ordinarios un modelo lineal que explique
#la variable Sales, en funcion otras
#realizamos la regresion multiple, donde ponemos a las variables,
#age, advertising y education para
##explicar la variacion de las ventas de sillitas
modelo <- lm(Sales ~ Age + Advertising + Education , data = chairs_net )
#usamos nuevamente la funcion lm, pero ahora sumaremos las variables explicativas
summary(modelo) # vemos el resultado
```

```
##
## Call:
## lm(formula = Sales ~ Age + Advertising + Education, data = chairs_net)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.646 -1.843 -0.025  1.633  7.550
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.948734   0.835016  10.717  < 2e-16 ***
## Age         -0.036884   0.008104  -4.551 7.43e-06 ***
## Advertising  0.139307   0.019752   7.053 9.90e-12 ***
## Education   -0.027461   0.050044  -0.549    0.584
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 2.424 on 339 degrees of freedom
## Multiple R-squared:  0.1687, Adjusted R-squared:  0.1614
## F-statistic: 22.94 on 3 and 339 DF,  p-value: 1.526e-13
```

```
#individualmente el efecto de cada variable se puede ver
#en estimate donde cada una de las pendientes
#dice que si se mantiente constante el resto de variables ,
#esta por cada unidad que aumenta
#la variable estudiada varia en tantas unidades como marca la pendiente
#en este caso
#age, por cada unidad que aumenta age las ventas de siilitas disminuyen un 0.036
#advetising, por cada unidad que aumenta los anuncios ls ventas aumentan 0.139
#education por cada unidad que aumenta la  el precio disminuye 0.027
#el r^2 explica la variabilidad del modelo, por lo que a mas
#variables mayor sera el valor de R^2
#en este caso vemos que explica el 16 por ciento de la variabilidad
# R^2 ajustado introduce una penalizacion al valor de R^2 por cada variable
#introducida
#tambien vemos que explica el 16 por ciento. lo que nos da un modelo
#de muy baja aceptacion
#vemos que el p value es significativo (menor que 1.5 e-13 ) por lo que
#se acepta que el modelo no es por azar
# vemos que individualmente todos los p value llamados Pr,
#son tambien muy bajos, todos con tres asteriscos menos education
# que significa segun la leyenda un numero considerado 0
```

realizamos un modelo mucho mas completo con todas las variables disponibles

```
#aplicamos el modelo de regresion lineal multiple
##realizamos lm
modelo_completo<- lm(Sales ~ Age + Population + Education + ShelveLoc + Advertising
                     + US + Urban+ Price + CompPrice, data = chairs_net )
summary(modelo_completo)
```

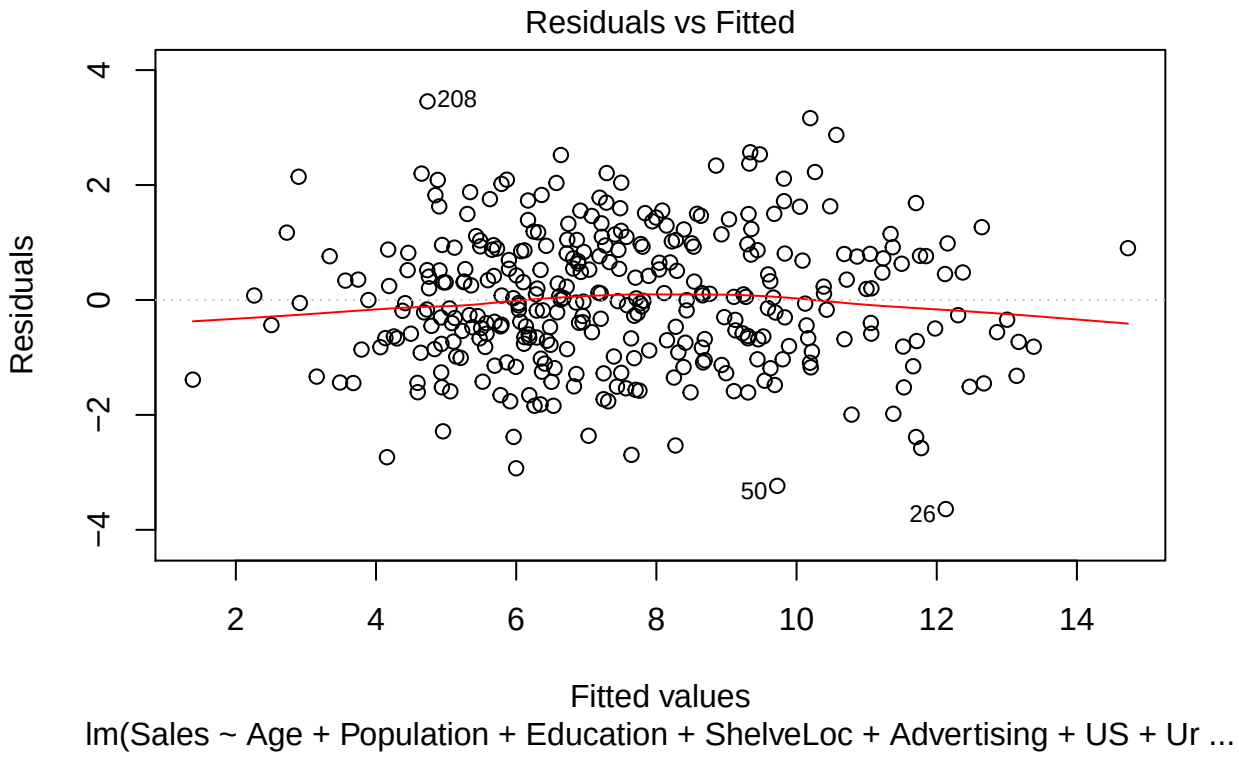
```
##
## Call:
## lm(formula = Sales ~ Age + Population + Education + ShelveLoc +
##     Advertising + US + Urban + Price + CompPrice, data = chairs_net)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6391 -0.7720 -0.0200  0.8421  3.4553
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      7.380e+00  6.501e-01  11.353   <2e-16 ***
## Age             -4.481e-02  4.039e-03 -11.096   <2e-16 ***
## Population       9.722e-06  4.646e-04   0.021    0.983
## Education       -2.223e-02  2.493e-02  -0.892    0.373
## ShelveLocMedium  1.899e+00  1.610e-01  11.797   <2e-16 ***
## ShelveLocGood    4.403e+00  1.957e-01  22.498   <2e-16 ***
## Advertising      1.246e-01  1.391e-02   8.960   <2e-16 ***
```

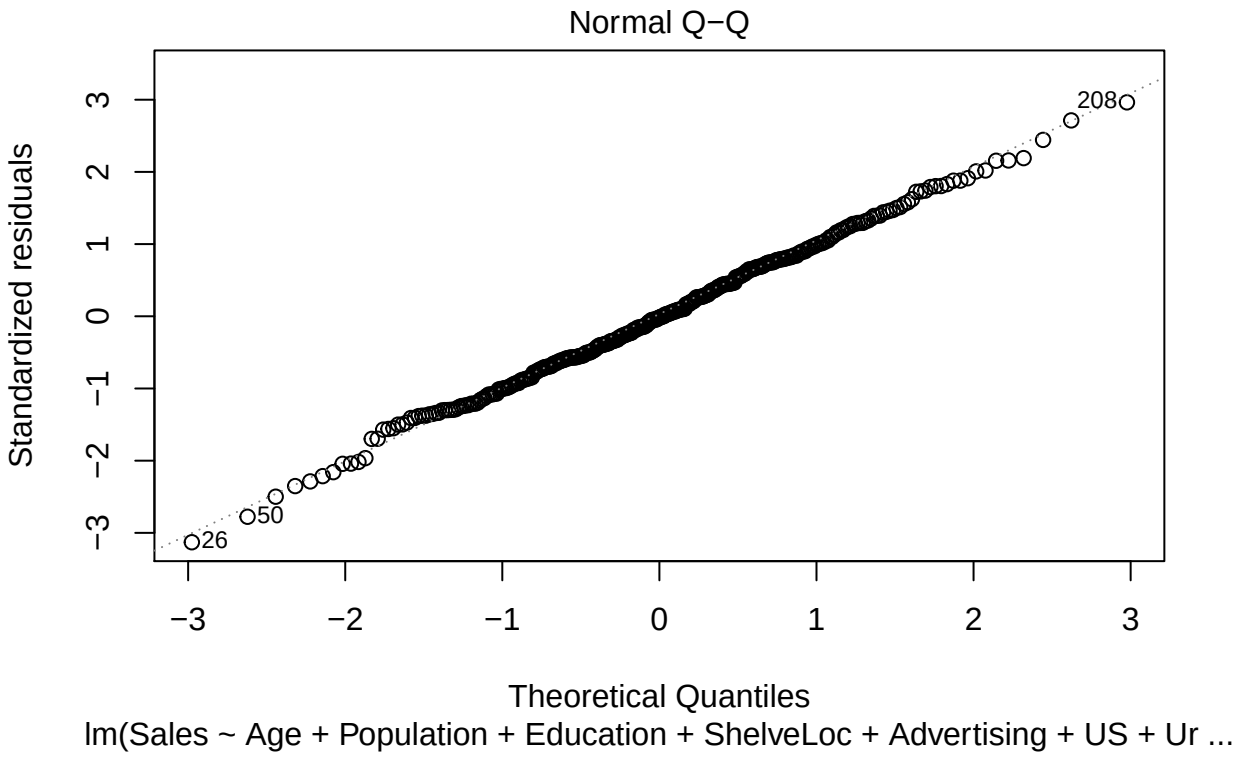
```
## USYes            1.138e-01  2.616e-01   0.435    0.664
## UrbanYes         1.285e-01  1.428e-01   0.900    0.369
## Price           -9.904e-02  3.920e-03 -25.267   <2e-16 ***
## CompPrice        9.015e-02  5.877e-03  15.339   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.19 on 332 degrees of freedom
## Multiple R-squared:  0.8038, Adjusted R-squared:  0.7979
## F-statistic: 136.1 on 10 and 332 DF,  p-value: < 2.2e-16
```

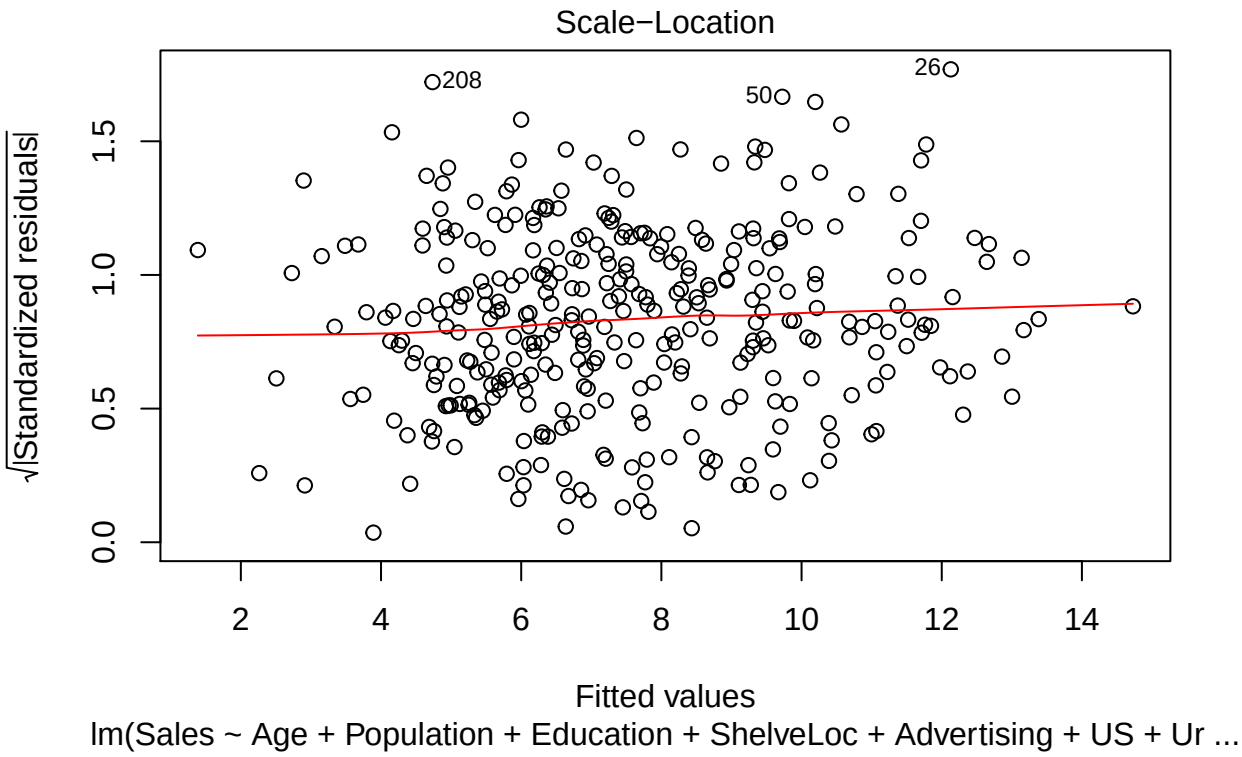
```
##vemos que la r^2 ajustada realmente precide el modelo, ya que explica el 79 por ciento de las variaci
#ademas de ver que la contribucion de shelveLoc es fundamental para ver las ventas
#por lo que es la variable mas significativa que tenemos
#como el p value es muy bajo podemos afirmar que el modelo es aceptable
```

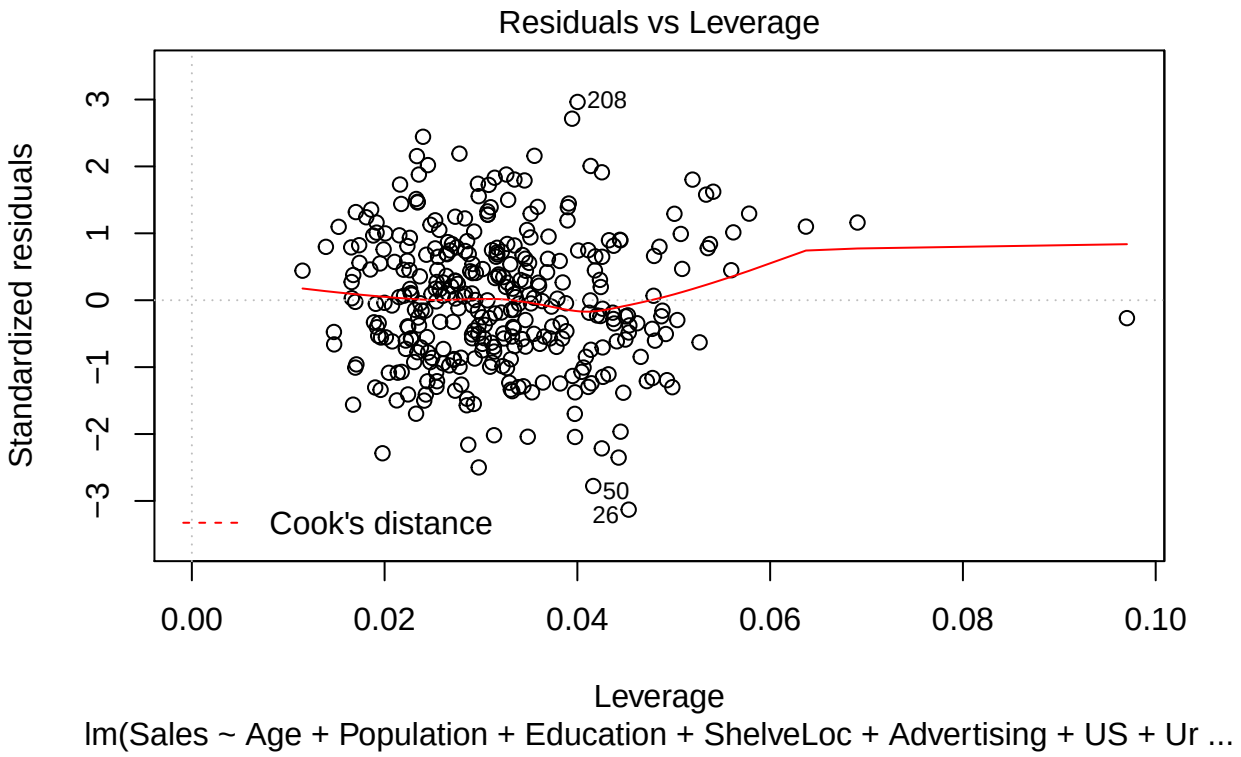
5 Representacion grafica

```
plot(modelo_completo)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(Sales ~ Age + Population + Education + ShelveLoc + Advertising + US + Ur ...
208
50
26
−3
−2
−1
0
1
2
3


Scale−Location
208
50
26
√|Standardized residuals|
0.0
0.5
1.0
1.5
2
4
6
8
10
12
14
Fitted values
lm(Sales ~ Age + Population + Education + ShelveLoc + Advertising + US + Ur ...




```
confint(lm(formula = Sales ~ Age + Population + Education + ShelveLoc + Advertising
                    + US + Urban+ Price + CompPrice , data = chairs_net))
```

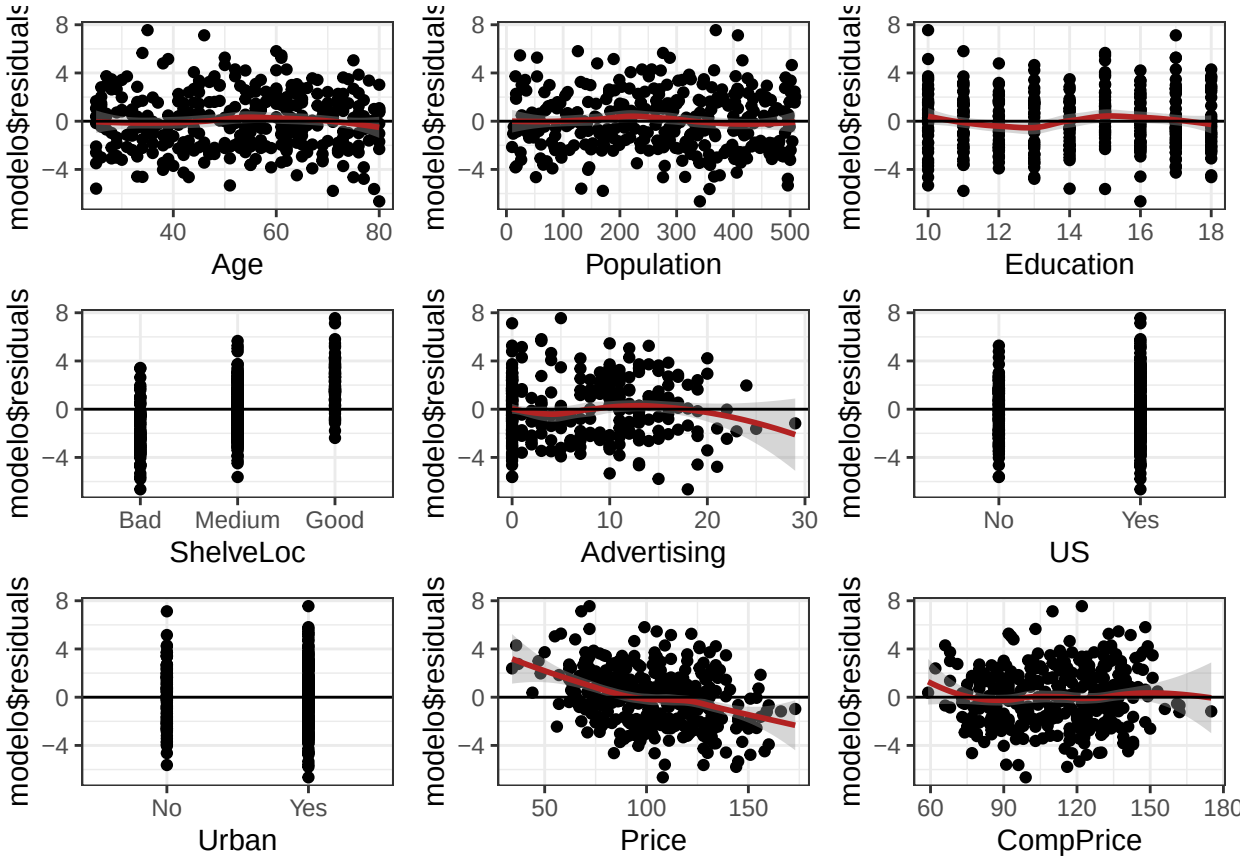
```
##                         2.5 %        97.5 %
## (Intercept)      6.1016158684  8.6592102445
## Age             -0.0527557556 -0.0368670941
## Population      -0.0009042842  0.0009237292
## Education       -0.0712791107  0.0268181110
## ShelveLocMedium  1.5825570861  2.2159260411
## ShelveLocGood    4.0176365795  4.7875085632
## Advertising      0.0972409718  0.1519477437
## USYes           -0.4008220990  0.6284469133
## UrbanYes        -0.1524057318  0.4094879628
## Price           -0.1067510973 -0.0913298179
## CompPrice        0.0785862754  0.1017074271
```

```
#realizamos un diagrama de dispersion entre cada una de las variables
#explicativas y los residuos
#si la distribucion es lineal los residuos deben distribuirse en torno
#a 0 con variabilidad constante
#en el eje x, lo que ocurre con las tres varaibles estudiadas
plot1 <- ggplot(data = chairs_net, aes(Age, modelo$residuals)) +
    geom_point() + geom_smooth(color = "firebrick") + geom_hline(yintercept = 0) +
    theme_bw()
plot2 <- ggplot(data = chairs_net, aes(Population, modelo$residuals)) +
```

```
    geom_point() + geom_smooth(color = "firebrick")+ geom_hline(yintercept = 0) +
    theme_bw()
plot3 <- ggplot(data = chairs_net, aes(Education, modelo$residuals)) +
    geom_point() + geom_smooth(color = "firebrick") + geom_hline(yintercept = 0) +
    theme_bw()
plot4 <- ggplot(data = chairs_net, aes(ShelveLoc, modelo$residuals)) +
    geom_point() + geom_smooth(color = "firebrick") + geom_hline(yintercept = 0) +
    theme_bw()
plot5 <- ggplot(data = chairs_net, aes(Advertising, modelo$residuals)) +
    geom_point() + geom_smooth(color = "firebrick")+ geom_hline(yintercept = 0) +
    theme_bw()
plot6 <- ggplot(data = chairs_net, aes(US, modelo$residuals)) +
    geom_point() + geom_smooth(color = "firebrick") + geom_hline(yintercept = 0) +
    theme_bw()
plot7 <- ggplot(data = chairs_net, aes(Urban, modelo$residuals)) +
    geom_point() + geom_smooth(color = "firebrick") + geom_hline(yintercept = 0) +
    theme_bw()
plot8 <- ggplot(data = chairs_net, aes(Price, modelo$residuals)) +
    geom_point() + geom_smooth(color = "firebrick")+ geom_hline(yintercept = 0) +
    theme_bw()
plot9 <- ggplot(data = chairs_net, aes(CompPrice, modelo$residuals)) +
    geom_point() + geom_smooth(color = "firebrick") + geom_hline(yintercept = 0) +
    theme_bw()

grid.arrange(plot1, plot2, plot3,plot4, plot5, plot6,plot7, plot8, plot9)
```

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



6 Conclusiones Vemos que ShelveLoc que es el tipo de calidad del agarre o abrochado , es lo mas decisivo a la hora de las ventas de sillitas, vemos tambien que la variable sales es normal,

las ventas en US son mayores que las ventas fuera y no hay distincion entre las ventas en sitios urbanos o rurales

Por desgracia como he comentado anteriormente mi compañera Mariana Tolivar, no ha podido realizar el trabajo por estar muy ocupada, por lo que yo he debido realizar tanto la investigación previa, la redacción de las respuestas y el desarrollo del código

Contribuciones Firma ..... Investigación previa R.H Redacción de las respuestas R.H Desarrollo codigo R.H