Limpieza y Análisis de Datos

Diciembre 2020

Contents

1 - DESCRIPCIÓN ACTIVIDAD

El objetivo de esta actividad es el tratamiento de un dataset, que puede ser el creado en la práctica 1 o bien estar disponible en Kaggle. En nuestro caso se trata de un dataset disponible en ***https://kaggle.com/https://www.kaggle.com/jmmvutu/summer-products-and-sales-in-ecommerce-wish***, y contiene información sobre las ventas de productos de Verano en la plataforma ecomerce **Wish**

1.1 - OBJETIVOS

- Aprender a aplicar los conocimientos adquiridos y su capacidad de resolución de problemas en entornos nuevos o poco conocidos dentro de contextos más amplios o multidisciplinares.
- Saber identificar los datos relevantes y los tratamientos necesarios (integración, limpieza y validación) para llevar a cabo un proyecto analítico.
- Aprender a analizar los datos adecuadamente para abordar la información contenida en los datos.
- Identificar la mejor representación de los resultados para aportar conclusiones sobre el problema planteado en el proceso analítico.
- Actuar con los principios éticos y legales relacionados con la manipulación de datos en función del ámbito de aplicación.
- Desarrollar las habilidades de aprendizaje que permita continuar estudiando de un modo que tendrá que ser en gran medida autodirigido o autónomo.
- Desarrollar la capacidad de búsqueda, gestión y uso de información y recursos en el ámbito de la ciencia de datos.

1.2 - COMPETENCIAS

En esta práctica se desarrollan las siguientes competencias del Master de Data Science:

- Capacidad de analizar un problema en el nivel de abstracción adecuado a cada situación y aplicar las habilidades y conocimientos adquiridos para abordarlo y resolverlo.
- Capacidad para aplicar las técnicas específicas de tratamiento de datos (integración, transformación, limpieza y validación) para su posterior análisis.

2 - RESOLUCIÓN

2.1 - DESCRIPCIÓN DEL DATASET / IMPORTANCIA

El conjunto de datos contiene listados de productos, así como calificaciones de productos y rendimiento de ventas obtenidos de la plataforma Wish si se escribe "verano" en el campo de búsqueda de dicha plataforma.

El dataset está formado por 43 características (columnas) que presentan 1575 sucesos (filas o registros), correspondientes a productos disponibles, ratios de venta, etc.:

title

Title for localized for european countries. May be the same as title_orig if the seller did not offer a translation

title_orig

Original english title of the product

price

price you would pay to get the product

retail_price

reference price for similar articles on the market, or in other stores/places. Used by the seller to indicate a regular value or

currency_buyer

currency of the prices

units_sold

Number of units sold. Lower bound approximation by steps

uses_ad_boosts

Whether the seller paid to boost his product within the platform (highlighting, better placement or whatever)

rating

Mean product rating

rating_count

Total number of ratings of the product

rating_five_count

Number of 5-star ratings

rating_four_count

Number of 4-star ratings

rating_three_count

Number of 3-star ratings

rating_two_count

Number of 2-star ratings

rating_one_count

Number of 1-star ratings

badges_count

Number of badges the product or the seller have

badge_local_product

A badge that denotes the product is a local product. Conditions may vary (being produced locally, or something else). Some

badge_product_quality

Badge awarded when many buyers consistently gave good evaluations 1 means Yes, has the badge

badge_fast_shipping

Badge awarded when this product's order is consistently shipped rapidly

product_color

Product's main color

tags

tags set by the seller

product_variation_size_id

One of the available size variation for this product

product_variation_inventory

Inventory the seller has. Max allowed quantity is 50

shipping_option_name
shipping_option_price

shipping price

shipping_is_express

whether the shipping is express or not. 1 for True

countries_shipped_to

Number of countries this product is shipped to. Sellers may choose to limit where they ship a product to

inventory_total

Total inventory for all the product's variations (size/color variations for instance)

has_urgency_banner

whether there was an urgency banner with an urgency

urgency_text

A text banner that appear over some products in the search results.

origin_country

merchant_title

Merchant's displayed name (show in the UI as the seller's shop name)

merchant_name

Merchant's canonical name. A name not shown publicly. Used by the website under the hood as a canonical name.

merchant_info_subtitle

The subtitle text as shown on a seller's info section to the user. (raw, not preprocessed).

merchant_rating_count

Number of ratings of this seller

merchant_rating

merchant's rating

merchant_id

merchant unique id

merchant_has_profile_picture

Convenience boolean that says whether there is a `merchant_profile_picture` url

merchant_profile_picture

Custom profile picture of the seller (if the seller has one). Empty otherwise.

product_url

url to the product page. You may need to login to access it

product_picture
product_id

product identifier. You can use this key to remove duplicate entries if you're not interested in studying them.

theme

the search term used in the search bar of the website to get these search results.

theme crawl_month

meta: for info only.

La información contenida en el dataset es interesante, ya que proporciona multitud de datos relacionados con los productos veraniegos que se venden en la plataforma. Podríamos considerar analizar la información desde perspectivas como las siguientes:

- Intentar validar la idea establecida de la sensibilidad humana a las caídas de precios (precio con descuento en comparación con el precio minorista original)
- Buscar las mejores categorías de productos para saber qué se vende mejor
- Comprobar si se venden los productos malos. Comprobar que hay de la relación entre la calidad de un producto (calificaciones) y su éxito. ¿El precio influye en esto? ...

A partir de este conjunto de datos, se plantea la problemática de determinar qué variables influyen más, y de que forma, sobre el precio del producto. También plantearemos algunas pruebas de contrastes de hipotésis, para confirmar o desmentir hechos que planteemos una vez analizados los datos y modelos de regresión para ver como ser relacionan las variables que consideremos más interesantes para conseguir nuestro objetivo.

Este análisis puede ser de gran utilidad, ya que puede ayudar a la plataforma a proporcionar información a los comerciantes sobre que parametrización de las ofertas es la más adecuada para incrementar sus ventas y fomentar el uso de la plataforma, en base al feedback proporcionado por los usuarios finales.

Trataremos tamibén de determinar que relación hay entre ventas de tallas grandes/pequeñas en relación al país de origen.

2.2 - INTEGRACIÓN Y SELECCIÓN DE DATOS

Una vez definido el objetivo, creemos que las características más relevantes a considerar inicialmente son:

price, retail_price, units_sold, uses_ad_boosts, rating, rating_count, rating_five_count,rating_four_count, rating_three_count,rating_two_count,rating_one_count,badges_count, badge_local_product,badge_product_quality, badge_fast_shipping,Tags,product_color,product_variation_inventory, shipping_is_express,countries_shipped_to,inventor merchant_rating,product_variation_size_id,origin_country

2.3 - LIMPIEZA DE LOS DATOS

Se realiza una inspección prelminar del archivo mediante Excel, donde, de entrada, no se observan valores vacíos, ni otro tipo de información que pueda ser problemática. El archivo csv viene separado por comas.

Hacemos la carga de las librerías necesarias:

```
# Lectura de los datos


SalesSummer <- read.csv("spwrap_2020_08.csv",header = TRUE)
```

```
# Tipos de datos asignados a cada campo

sapply(SalesSummer, function(x) class(x))
```

```
##                         title                    title_orig
##                   "character"                   "character"
##                         price                  retail_price
##                     "numeric"                     "integer"
##                currency_buyer                    units_sold
##                     "character"                     "integer"
##                uses_ad_boosts                        rating
##                     "integer"                     "numeric"
##                  rating_count             rating_five_count
##                     "integer"                     "integer"
##            rating_four_count           rating_three_count
##                     "integer"                     "integer"
##             rating_two_count             rating_one_count
##                     "integer"                     "integer"
##                   badges_count          badge_local_product
##                     "integer"                     "integer"
##        badge_product_quality          badge_fast_shipping
##                     "integer"                     "integer"
##                         tags                 product_color
##                   "character"                   "character"
##    product_variation_size_id  product_variation_inventory
##                   "character"                     "integer"
##         shipping_option_name        shipping_option_price
##                   "character"                     "integer"
##          shipping_is_express         countries_shipped_to
##                     "integer"                     "integer"
##              inventory_total           has_urgency_banner
```

```
##                   "integer"                   "integer"
##                urgency_text               origin_country
##                 "character"                   "character"
##              merchant_title                merchant_name
##                 "character"                   "character"
##       merchant_info_subtitle        merchant_rating_count
##                 "character"                     "integer"
##             merchant_rating                   merchant_id
##                   "numeric"                   "character"
## merchant_has_profile_picture     merchant_profile_picture
##                   "integer"                   "character"
##                 product_url              product_picture
##                 "character"                   "character"
##                  product_id                        theme
##                 "character"                   "character"
##                 crawl_month
##                 "character"
```

Comprobamos que los tipos proporcinados para cada columna coinciden con los del dataset.

2.3.1 - Selección de los datos de interes

Siguiendo el criterio establecido en el apartado 2.2, vamos a seleccionar del dataset las columnas: price, retail_price, units_sold, uses_ad_boosts, rating, rating_count, rating_five_count,rating_four_count, rating_three_count,rating_two_count,rating_one_count,badges_count,badge_local_product, badge_product_quality, badge_fast_shipping,tags,product_color,product_variation_inventory, shipping_is_express,countries_shipped_to,inventor

has_urgency_banner parece una variable entera interesante(0,1), pero comprobamos que hay 1100 registros con un valor NA y el resto es siempre 1, con lo que resulta inviable su uso al no poder asignar un valor de forma coherente a dicho registros

En una primera inspección detectamos valores NA, y algunas filas sin ningún valor asignado en las variables: product_color,product_variation_size_id,origin_country

2.3.2 - Ceros y elementos vacíos

Vamos a comprobar si tenemos ceros y/o elementos vacíos

```
# Comprobamos valores NA


sapply(SalesSummerObj, function(x) sum(is.na(x)))
```

```
##                      price                retail_price
##                          0                           0
##                 units_sold              uses_ad_boosts
##                          0                           0
##                     rating                rating_count
##                          0                           0
##           rating_five_count           rating_four_count
##                         45                          45
##          rating_three_count            rating_two_count
##                         45                          45
```

```
##            rating_one_count                badges_count
##                          45                           0
##         badge_local_product       badge_product_quality
##                           0                           0
##         badge_fast_shipping                        tags
##                           0                           0
##               product_color product_variation_inventory
##                           0                           0
##         shipping_is_express        countries_shipped_to
##                           0                           0
##             inventory_total             merchant_rating
##                           0                           0
##   product_variation_size_id              origin_country
##                           0                           0
```

```
# Comprobamos valores nulos


sapply(SalesSummerObj, function(x) sum(is.null(x)))
```

```
##                       price                retail_price
##                           0                           0
##                  units_sold              uses_ad_boosts
##                           0                           0
##                      rating                rating_count
##                           0                           0
##           rating_five_count           rating_four_count
##                           0                           0
##          rating_three_count            rating_two_count
##                           0                           0
##            rating_one_count                badges_count
##                           0                           0
##         badge_local_product       badge_product_quality
##                           0                           0
##         badge_fast_shipping                        tags
##                           0                           0
##               product_color product_variation_inventory
##                           0                           0
##         shipping_is_express        countries_shipped_to
##                           0                           0
##             inventory_total             merchant_rating
##                           0                           0
##   product_variation_size_id              origin_country
##                           0                           0
```

No tenemos valores nulos en las variables a contemplar.

Los 45 valores NA detectados en las variables rating_five_count,rating_four_count,rating_three_count,rating_two_count, rating_one_count, se deben al valor 0 en la variable rating_count. Es decir no hay desglose entre distintos tipos de rating si el contador total es cero. El rating está calculado a partir del rating_count y la distribución de ratings:

rating = rating5 * 5 + rating4 * 4 + rating3 * 3 + rating2 * 2 + rating1 / rating_count

A efectos de cálculo sustituimos los valores NA por cero

```
SalesSummerObj$rating_five_count[is.na(SalesSummerObj$rating_five_count)] <- 0
SalesSummerObj$rating_four_count[is.na(SalesSummerObj$rating_four_count)] <- 0
SalesSummerObj$rating_three_count[is.na(SalesSummerObj$rating_three_count)] <- 0
SalesSummerObj$rating_two_count[is.na(SalesSummerObj$rating_two_count)] <- 0
SalesSummerObj$rating_one_count[is.na(SalesSummerObj$rating_one_count)] <- 0
```

La variable product_color tiene algunos valores sin información. Vamos a modificar esos valores asignando un string "No color".

```
SalesSummerObj$product_color[SalesSummerObj$product_color==""] <- "no color"
```

La variable product_color tiene algunos colores iguales pero representados de forma diferente, y que vamos a homogeneizar, para después factorizarlos correctamente:

```
SalesSummerObj$product_color[SalesSummerObj$product_color=="Army green" |
                             SalesSummerObj$product_color=="armygreen" ] <- "army green"
SalesSummerObj$product_color[SalesSummerObj$product_color=="wine red"  ] <- "winered"
SalesSummerObj$product_color[SalesSummerObj$product_color=="RED"  ] <- "red"
SalesSummerObj$product_color[SalesSummerObj$product_color=="Rose red"  ] <- "rosered"
SalesSummerObj$product_color[SalesSummerObj$product_color=="White"  ] <- "white"
SalesSummerObj$product_color[SalesSummerObj$product_color=="Pink"  ] <- "pink"
```

La variable origin_country tiene algunos valores sin información. Vamos a modificar esos valores asignando un string "NC".

```
SalesSummerObj$origin_country[SalesSummerObj$origin_country==""] <- "NC"
```

La variable product_variation_size_id tiene algunos valores sin información. Vamos a modificar esos valores asignando un string "No size".

```
SalesSummerObj$product_variation_size_id[SalesSummerObj$product_variation_size_id ==""] <- "No size"
```

2.3.3 - Identificación y tratamiento de outliers

Un outlier es una observación anormal y extrema en una muestra estadística o serie temporal de datos, que puede afectar potencialmente a la estimación de los parámetros del mismo.

```
summary(SalesSummerObj)
```

```
##      price          retail_price      units_sold     uses_ad_boosts
##  Min.   : 1.000   Min.   :  1.00   Min.   :     1   Min.   :0.0000
##  1st Qu.: 5.810   1st Qu.:  7.00   1st Qu.:   100   1st Qu.:0.0000
##  Median : 8.000   Median : 10.00   Median :  1000   Median :0.0000
##  Mean   : 8.325   Mean   : 23.29   Mean   :  4339   Mean   :0.4329
##  3rd Qu.:11.000   3rd Qu.: 26.00   3rd Qu.:  5000   3rd Qu.:1.0000
##  Max.   :49.000   Max.   :252.00   Max.   :100000   Max.   :1.0000
##      rating       rating_count     rating_five_count rating_four_count
##  Min.   :1.000   Min.   :    0.0   Min.   :    0.0   Min.   :   0.0
##  1st Qu.:3.550   1st Qu.:   24.0   1st Qu.:   10.0   1st Qu.:   4.0
##  Median :3.850   Median :  150.0   Median :   72.0   Median :  29.0
```

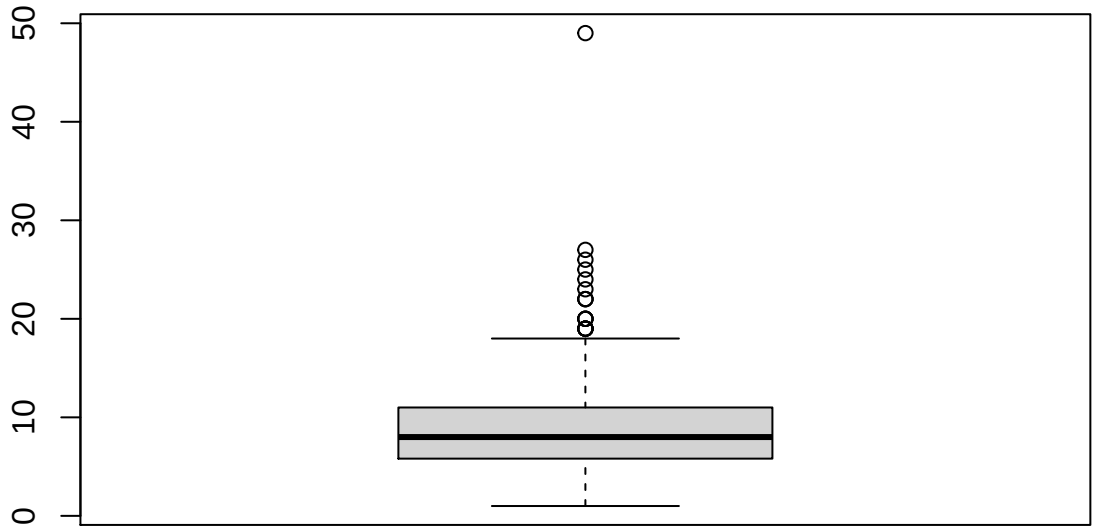
```
##  Mean   :3.821   Mean   :  889.7   Mean   :  429.6   Mean   : 174.5
##  3rd Qu.:4.110   3rd Qu.:  855.0   3rd Qu.:  394.0   3rd Qu.: 163.0
##  Max.   :5.000   Max.   :20744.0   Max.   :11548.0   Max.   :4152.0
##  rating_three_count rating_two_count  rating_one_count  badges_count
##  Min.   :   0.0     Min.   :   0.00   Min.   :   0     Min.   :0.0000
##  1st Qu.:   3.0     1st Qu.:   1.00   1st Qu.:   3     1st Qu.:0.0000
##  Median :  22.0     Median :  10.00   Median :  18     Median :0.0000
##  Mean   : 130.7     Mean   :  61.89   Mean   :  93     Mean   :0.1055
##  3rd Qu.: 121.0     3rd Qu.:  59.00   3rd Qu.:  90     3rd Qu.:0.0000
##  Max.   :3658.0     Max.   :2003.00   Max.   :2789     Max.   :3.0000
##  badge_local_product badge_product_quality badge_fast_shipping
##  Min.   :0.00000     Min.   :0.00000       Min.   :0.00000
##  1st Qu.:0.00000     1st Qu.:0.00000       1st Qu.:0.00000
##  Median :0.00000     Median :0.00000       Median :0.00000
##  Mean   :0.01844     Mean   :0.07438       Mean   :0.01271
##  3rd Qu.:0.00000     3rd Qu.:0.00000       3rd Qu.:0.00000
##  Max.   :1.00000     Max.   :1.00000       Max.   :1.00000
##      tags           product_color      product_variation_inventory
##  Length:1573        Length:1573        Min.   : 1.00
##  Class :character   Class :character   1st Qu.: 6.00
##  Mode  :character   Mode  :character   Median :50.00
##                                        Mean   :33.08
##                                        3rd Qu.:50.00
##                                        Max.   :50.00
##  shipping_is_express countries_shipped_to inventory_total merchant_rating
##  Min.   :0.000000    Min.   :  6.00       Min.   : 1.00   Min.   :2.333
##  1st Qu.:0.000000    1st Qu.: 31.00       1st Qu.:50.00   1st Qu.:3.917
##  Median :0.000000    Median : 40.00       Median :50.00   Median :4.041
##  Mean   :0.002543    Mean   : 40.46       Mean   :49.82   Mean   :4.032
##  3rd Qu.:0.000000    3rd Qu.: 43.00       3rd Qu.:50.00   3rd Qu.:4.162
##  Max.   :1.000000    Max.   :140.00       Max.   :50.00   Max.   :5.000
##  product_variation_size_id origin_country
##  Length:1573               Length:1573
##  Class :character          Class :character
##  Mode  :character          Mode  :character
##
##
##
```

```
boxplot.stats(SalesSummerObj$price)$out
```

```
##  [1] 20 22 19 19 19 20 24 22 49 19 23 22 20 25 19 26 20 19 27
```

Vemos un único valor significativamente elevado (49).Vamos a considerarlo como outlier y eliminamos el registro del conjunto de datos.

```
boxplot(SalesSummerObj$price)
```



Vamos a considerarlo como outlier y eliminamos el registro que lo contiene del conjunto de datos.

```
SalesSummerObj <- SalesSummerObj[!(SalesSummerObj$price == 49),]
```

```
boxplot.stats(SalesSummerObj$retail_price)$out
```

```
##   [1]  84  81  76  81  68  56  60  56  68  67  92  92  65  67  85  59  56  76
##  [19]  56  84  76 115  89  84 145  59 169  56  76  65  84 101  89  56  59  84
##  [37]  59  88 118  57 104  81  89  60 118  75  84  59  58  75 134 115  59 106
##  [55] 108 152 106  85  72 159 108 159  68  76  56 140 168 168  85  81  85  75
##  [73]  59  59  84  76  68  59  93  84 122  85  75  72  84  86 127  70 140 159
##  [91] 159 127 100 126  97  58 250  85  85  68 118  92 109  85  55  84  58  84
## [109]  84 111  65  67  84  59  56  84  93  59  69  85  85  56 142  86  65  58
## [127]  66  76 102  59  83 105  59  59  68  84  56  55  84  58 152  56  85  85
## [145]  84 110 102  75  68  84  68  59 168 252 168  85  76  65  65 140 102  59
## [163]  67  68 102  60  84  84  85 102  59 135  65 107  93  59 252  83  68  85
## [181]  75  68  75 139  84  84  84  59 108  76  84  59  84  59  68 108 168  88
## [199]  76  76  64  84  87  72  76  92 134  56
```

Idem caso vaiable price.

```
boxplot.stats(SalesSummerObj$units_sold)$out
```

```
##   [1]  20000  50000  20000 100000  20000  20000  20000  50000  50000  20000
##  [11]  20000 100000  20000  20000  20000  20000  20000  20000  20000  20000
##  [21] 100000  20000 100000  20000  20000  20000  20000  20000  20000  50000
##  [31]  20000  20000  20000  50000  50000  20000  20000  20000  20000  20000
##  [41]  20000  20000  20000  20000  50000  20000  50000  20000  20000  20000
##  [51]  20000  20000  20000  20000  20000  20000  20000  20000  20000  20000
##  [61]  20000  20000  20000  20000  20000  20000  20000  20000  20000  20000
##  [71]  50000  20000  20000  20000  20000  50000  50000  20000  50000  50000
##  [81]  20000  20000  20000  20000  20000  20000 100000  20000  50000  20000
##  [91]  20000  20000  20000  20000  20000  20000  20000 100000  20000  50000
## [101]  50000  20000  20000  20000  20000  20000  20000  20000  20000  20000
## [111]  20000  20000  20000  20000  20000  50000  20000  20000  20000  20000
## [121]  20000  20000  20000  20000  20000  20000
```

Todos los valores se repiten y hablamos de unidades vendidas, por lo que no vamos a considerarlos como outliers

```
boxplot.stats(SalesSummerObj$rating)$out
```

```
##   [1] 5.00 1.50 5.00 5.00 2.61 5.00 5.00 5.00 5.00 2.50 2.67 2.00 5.00 5.00 5.00
##  [16] 2.00 5.00 2.00 5.00 5.00 5.00 5.00 2.00 5.00 5.00 5.00 1.00 5.00 2.50 5.00
##  [31] 5.00 5.00 2.67 5.00 2.61 2.00 2.67 5.00 2.25 5.00 5.00 5.00 2.00 5.00 2.00
##  [46] 5.00 2.67 5.00 2.57 5.00 2.67 5.00 1.00 5.00 5.00 5.00 5.00 2.33 5.00 5.00
##  [61] 5.00 5.00 5.00 5.00 5.00 2.57 5.00 2.00 2.44 5.00 5.00 5.00 5.00 5.00 5.00
##  [76] 5.00 5.00 5.00 5.00 5.00 5.00 2.67 5.00 5.00 5.00 5.00 5.00 5.00 2.00 5.00
##  [91] 1.50 2.33 5.00 1.00 5.00 5.00 5.00 5.00 5.00 2.44 5.00 2.50
```

Hablamos de un rating que va entre 1 y 5, y que ha sido calculado de origen a partir de las otras variable rating. No vamos a efectuar cambios sobre ellos

```
boxplot.stats(SalesSummerObj$rating_five_count)$out
```

```
##   [1]  2269  3172   984  8290  2456  1437  1781  3111  1467  1220  1162 11548
##  [13]  2054  1635 11184  1726  1427  1782  4640  3906  1099  4468  1695  2341
##  [25]  2245  1314  1070  1074  4663  1834  1064  1428  6862  2138  1352  2077
##  [37]  1646  1415   985  1056  1573  1158  2089  3367  1009  2418  2611   994
##  [49]  1123  3502  1352  2936  1511  3278  4299  3566  1138  4974  2050  1095
##  [61]  3598  2998  1245  1046  1201  1383  5065  4487  1141  1253  1813  2301
##  [73]  1950  3030  1964  2638  3029  3624  1328  1891  1032  1228  1184  2679
##  [85]   973  1375  2620  3689  1097  1506  1222  2091  1342   981  1417  2616
##  [97]  2246  1009  1223  1334  1974  1783  5479  1511  1220  1271  1958   981
## [109]  1044  3140  5077  1632  1269  5946  3689  1442  6060  7337  4640  1325
## [121]  1315  1582  1194  2720  1115  2090  1306  5355  2541  6325  1281  1789
## [133]  1646  1975  1089  2045  1898  1209  1260  2943  1530  3383  3360  1578
## [145]  1316  1464  6769  2089  4182  5274  4150  1337  1975  1232  5723  1253
## [157]  1141  1125   981  2077  1162  2867  1428  1056  1070   976  1736  1334
## [169]  1182   975  2011  4220  2184  1542  1124  7530  1491  1122  4130  1800
## [181]  1116  2940  4166  1312  2458  1311  3125  2107  1324  2635  1573  1919
```

```
boxplot.stats(SalesSummerObj$rating_four_count)$out
```

```
##   [1] 1027 1352  481  435 3483 1162  632  628 1172  607  433  488 3191  774  642
##  [16] 4152  654  614  565 1964 1326  449 1850  721  878 1059  460 2418  691  522
##  [31] 2836  667  574  783  632  448  419  546  545  901  539 1510  548  431  850
##  [46]  449  468 1523  636 1396  479 1411  417 1527 1389  523 2091  550  597 1821
##  [61] 1514  418  453  591  405 2342 1955 1162  939  776 1257 1029  851 1251 1733
##  [76]  706  631  552  545 1403  544  491 1456 1502  482  417  654  492  469  612
##  [91]  691  434  804 1011 1221  644  459  581  730 2331  479  433  675  434  526
## [106]  422 1357 1860  531  485 2952 1502  427  440  635 2562 2647 1964  521  482
## [121]  524  480 1171  445  852  463 2430  673 3006  538  459  564  793  648  509
## [136]  426  449  582 1022  500 1430 1065  562  487  713 3404  901 1317 1869 1941
## [151]  495  793  593 2701  470  445  783  542 1406  505  484  555  419  460  573
## [166]  406 1127 1795  717  573 3351  717  520  994  575 1130 1546  611  726  663
## [181] 1205 1226  488 1394  606  580
```

```
boxplot.stats(SalesSummerObj$rating_three_count)$out
```

```
##   [1] 1118  971  459  386 2951  853  610  374  650  558  344  365 1632  576  392
##  [16] 2919  568  452  332 1121  869  362 1277  416  607  999  373 1868  340  385
##  [31] 2131  637  443  446  324  305  320  375  374  370  650  410  301 1148  358
##  [46]  327  317  449  354 1145  397 1547  331  998  331  331 1177  942  521 1409
##  [61]  464 1607 1827  522  342  345 1964 1532 1071  572  428  884 1167  473  982
##  [76] 1531  611  367  394  329 1317  332  456 1235 1160  430  331  417  345  621
##  [91]  328  336  618  368  786  657 1098  419  339  566 1605  365  518  368  348
## [106]  317 1154 1185  355 2624 1160  379  401 2214 1643 1121  360  343  383  401
## [121]  820  576  305 1974  309 1998  365  389  682  354  334  392  662  836  500
## [136]  339  586 3658  650  857 1519 1495  682  622 1656  390  324  324  301  446
## [151]  559 1330  513  350  340  375  373  849 1098  605  448 3057  430  331  386
## [166]  409  364  712 1044  337  355  461  838 1189 1315  399  304
```

```
boxplot.stats(SalesSummerObj$rating_two_count)$out
```

```
##   [1]  644  490  206  158 1410  431  358  204  241  189  203  151  507  242  177
##  [16] 1174  324  180  493  343  154  583  230  562  181  842  171  901  255  227
##  [31]  192  156  190  159  181  287  169  162  469  212  416  185  970  184  431
##  [46]  223  194  527  352  324  655  222  808  149 1136  289  153  164  890  658
##  [61]  609  248  184  326  684  213  455  761  377  165  793  190  247  515  592
##  [76]  227  194  217  152  149  510  196  301  220  411  284  506  178  201  258
##  [91]  665  161  203  242  220  169  151  153  445  467  153 1310  592  168 1033
## [106]  623  493  152  263  180  223  408  278  157  960  916  397  226  258  335
## [121]  159  309 2003  287  429  684  731  397  327  751  233  191  170  192  339
## [136]  695  315  182  190  181  360  351  271  212 1736  178  184  173  148  295
## [151]  438  291  305  553  715
```

```
boxplot.stats(SalesSummerObj$rating_one_count)$out
```

```
##   [1] 1077  757  327  239 1846  577  515  222  335  426  299  318  566  329  224
##  [16] 1315  548  273  238  686  416  230  782  244  256  289  776  357  230  258
##  [31] 1271  224  231  247 1059  385  269  296  343  382  256  265  377  224  399
##  [46]  594  344  229  436  271 1315  327  617  344  316  882  440  232  710 1021
```

```
##  [61]  254  328 1137  277  334 1600  567  225  272 1147  705 1030  318  222  432
##  [76]  937  246  643  960  550 1404  368  418  683 1086  397  316  307  390  334
##  [91]  958  402  299  445  229  363  511  306  820  241  267  396  363  847  299
## [106]  368  363  289  316  244  349  414  618  287  264 1736 1086  261 1329  630
## [121]  686  272  234  253  661  384  600  341  233 1194 1243  233  520  354  446
## [136]  347  278  466  502 2559  377  790  776 1013  281  520  669 1210  401  322
## [151]  234  742  944  465  328  382  357  310  309  438  305 2789  327  226  236
## [166]  274  559  353  286  681  257  267 1029
```

Tampoco vamos a realizar alteraciones sobre los valores de los ratings 1-5

```
boxplot.stats(SalesSummerObj$product_variation_inventory)$out
```

```
## integer(0)
```

```
boxplot.stats(SalesSummerObj$inventory_total)$out
```

```
## [1] 40 36  1 30  9 24 37 38  2
```

Se trata de valores de inventario que no vamos a categorizar como outliers.

```
boxplot.stats(SalesSummerObj$merchant_rating)$out
```

```
##  [1] 3.507692 3.548387 3.298507 3.548387 3.186047 3.473684 3.534647 3.545024
##  [9] 3.409471 3.515748 3.034483 5.000000 2.941176 3.417722 3.409471 3.381868
## [17] 3.545024 3.038961 3.475584 3.515748 2.333333 3.464286 3.338290 3.381868
## [25] 4.577519 3.187500 3.514286 3.186047 3.187500 3.548387 3.367133 3.250000
## [33] 3.422535 3.475584 3.000000 3.545455 3.534647 3.545024
```

Se trata de un rating que va entre 2.333 y 5, no vamos a caterogizarlos como outliers

2.3.4 - Exportación de los datos preprocesados

Exportamos los datos preprocesados a un fichero .csv

```
# Exportación de los datos preprocesados a un fichero .csv

write.csv(SalesSummerObj,"spwrap_2020_08_data_clean.csv")
```

2.3.5 - Factorización y niveles de las variables cuantitativas

Vamos a factorizar la variable product_color.

```
# Convertimos en factor y vemos sus niveles


levels(factor(SalesSummerObj$product_color))
```

```
##  [1] "applegreen"          "apricot"             "army"
##  [4] "army green"          "beige"               "black"
##  [7] "Black"               "black & blue"        "black & green"
## [10] "black & stripe"      "black & white"       "black & yellow"
## [13] "blackwhite"          "blue"                "Blue"
## [16] "blue & pink"         "brown"               "brown & yellow"
## [19] "burgundy"            "camel"               "camouflage"
## [22] "claret"              "coffee"              "coolblack"
## [25] "coralred"            "darkblue"            "darkgreen"
## [28] "denimblue"           "dustypink"           "floral"
## [31] "fluorescentgreen"    "gold"                "gray"
## [34] "gray & white"        "green"               "grey"
## [37] "greysnakeskinprint"  "ivory"               "jasper"
## [40] "khaki"               "lakeblue"            "leopard"
## [43] "leopardprint"        "light green"         "lightblue"
## [46] "lightgray"           "lightgreen"          "lightgrey"
## [49] "lightkhaki"          "lightpink"           "lightpurple"
## [52] "lightred"            "lightyellow"         "mintgreen"
## [55] "multicolor"          "navy"                "navy blue"
## [58] "navyblue"            "navyblue & white"    "no color"
## [61] "nude"                "offblack"            "offwhite"
## [64] "orange"              "orange-red"          "orange & camouflage"
## [67] "pink"                "pink & black"        "pink & blue"
## [70] "pink & grey"         "pink & white"        "prussianblue"
## [73] "purple"              "rainbow"             "red"
## [76] "red & blue"          "rose"                "rosegold"
## [79] "rosered"             "silver"              "skyblue"
## [82] "star"                "tan"                 "violet"
## [85] "watermelonred"       "white"               "white & black"
## [88] "white & green"       "white & red"         "whitefloral"
## [91] "whitestripe"         "wine"                "winered"
## [94] "winered & yellow"    "yellow"
```

Factorizamos los valores para dicha variable

```
SalesSummerObj$product_color <- as.numeric(factor(SalesSummerObj$product_color))
```

2.4 - ANÁLISIS DE LOS DATOS

2.4.2 - Selección de grupos de datos

Seleccionamos un conjunto inicial de grupos de datos que nos pueden resultar interesantes de analizar y/o comparar.

Agrupación por utilización de anuncios uses_ad_boosts (0/1)

```
SalesSummerObj.uses.ad.boosts.cero <- SalesSummerObj %>% filter(uses_ad_boosts == "0")
```

```
SalesSummerObj.uses.ad.boosts.uno <- SalesSummerObj %>% filter(uses_ad_boosts == "1")
```

Agrupación por insignia local product

```
SalesSummerObj.badget.localproduct.cero <- SalesSummerObj %>% filter(badge_local_product== "0")
```

```
SalesSummerObj.badget.localproduct.uno <- SalesSummerObj %>% filter(badge_local_product== "1")
```

Agrupación por insignia product quality

```
SalesSummerObj.badget.productquality.cero <- SalesSummerObj %>% filter(badge_product_quality== "0")
```

```
SalesSummerObj.badget.productquality.uno <- SalesSummerObj %>% filter(badge_product_quality == "1")
```

Agrupación por insignia fast shipping

```
SalesSummerObj.badget.fastshipping.cero <- SalesSummerObj %>% filter(badge_fast_shipping == "0")
```

```
SalesSummerObj.badget.fastshipping.uno <- SalesSummerObj %>% filter(badge_fast_shipping == "1")
```

Agrupación por shipping express

```
SalesSummerObj.shipping.express.cero <- SalesSummerObj %>% filter(shipping_is_express == "0")
```

```
SalesSummerObj.shipping.express.uno <- SalesSummerObj %>% filter(shipping_is_express == "1")
```

Agrupación por intervalos de rating

rating <=1.5 -> Intervalo 1 rating >1.5 and < 2.5 -> Intervalo 2 rating >=2.5 and < 3.5 -> Intervalo 3 rating >=3.5 and < 4.5 -> Intervalo 4 rating >= 4.5 -> Intervalo 5

Para ello crearemos una variable rating_interval donde asignaremos el valor 1 a 5 dependiendo del rango de valores definidos:

```
SalesSummerObj <- cbind(SalesSummerObj,rating_interval=c(as.integer(0)))
```

```
SalesSummerObj$rating_interval[SalesSummerObj$rating <= 1.5 ] <- 1
SalesSummerObj$rating_interval[SalesSummerObj$rating > 1.5 & SalesSummerObj$rating < 2.5  ] <- 2
SalesSummerObj$rating_interval[SalesSummerObj$rating >=2.5 & SalesSummerObj$rating < 3.5  ] <- 3
SalesSummerObj$rating_interval[SalesSummerObj$rating >=3.5 & SalesSummerObj$rating < 4.5  ] <- 4
SalesSummerObj$rating_interval[SalesSummerObj$rating >=4.5] <- 5
```

```
SalesSummerObj.rating.interval.uno <- SalesSummerObj %>% filter(SalesSummerObj$rating_interval == "1")
```

```
SalesSummerObj.rating.interval.dos <- SalesSummerObj %>% filter(SalesSummerObj$rating_interval == "2")
```

```
SalesSummerObj.rating.interval.tres <- SalesSummerObj %>% filter(SalesSummerObj$rating_interval == "3")
```

```
SalesSummerObj.rating.interval.cuatro <- SalesSummerObj %>% filter(SalesSummerObj$rating_interval == "4
```

```
SalesSummerObj.rating.interval.cinco <- SalesSummerObj %>% filter(SalesSummerObj$rating_interval == "5"
```

*** Agrupación por tallas grandes y pequeñas ***

Añadimos una variable size_category inicializada con valor EC (Empty Category)

```
SalesSummerObj <- cbind(SalesSummerObj,size_category=c("EC"))
```

Seteamos la nueva variable en funcion de tallas pequeñas (SS) o grandes (HS)

```
SalesSummerObj$size_category[SalesSummerObj$product_variation_size_id == "XS" |
SalesSummerObj$product_variation_size_id == "" | SalesSummerObj$product_variation_size_id == "XXS" |
SalesSummerObj$product_variation_size_id == "XXXS" | SalesSummerObj$product_variation_size_id == "S"] <-

SalesSummerObj$size_category[SalesSummerObj$product_variation_size_id == "XL" |
SalesSummerObj$product_variation_size_id == "XXL" |SalesSummerObj$product_variation_size_id ==  "XXXL"
SalesSummerObj$product_variation_size_id == "XXXXL" | SalesSummerObj$product_variation_size_id == "XXXX
```

2.4.3 - Comprobación de normalidad y homogeneidad de la varianza

Para la comprobación de que los valores que toman nuestras variables cuantitativas provienen de una población distribuida normalmente, utilizaremos la prueba de normalidad de Shapiro.

Así, se comprueba que para que cada prueba se obtiene un p-valor superior al nivel de significación prefijado alfa = 0.05. Si esto se cumple, entonces se considera que variable en cuestión sigue una distribución normal

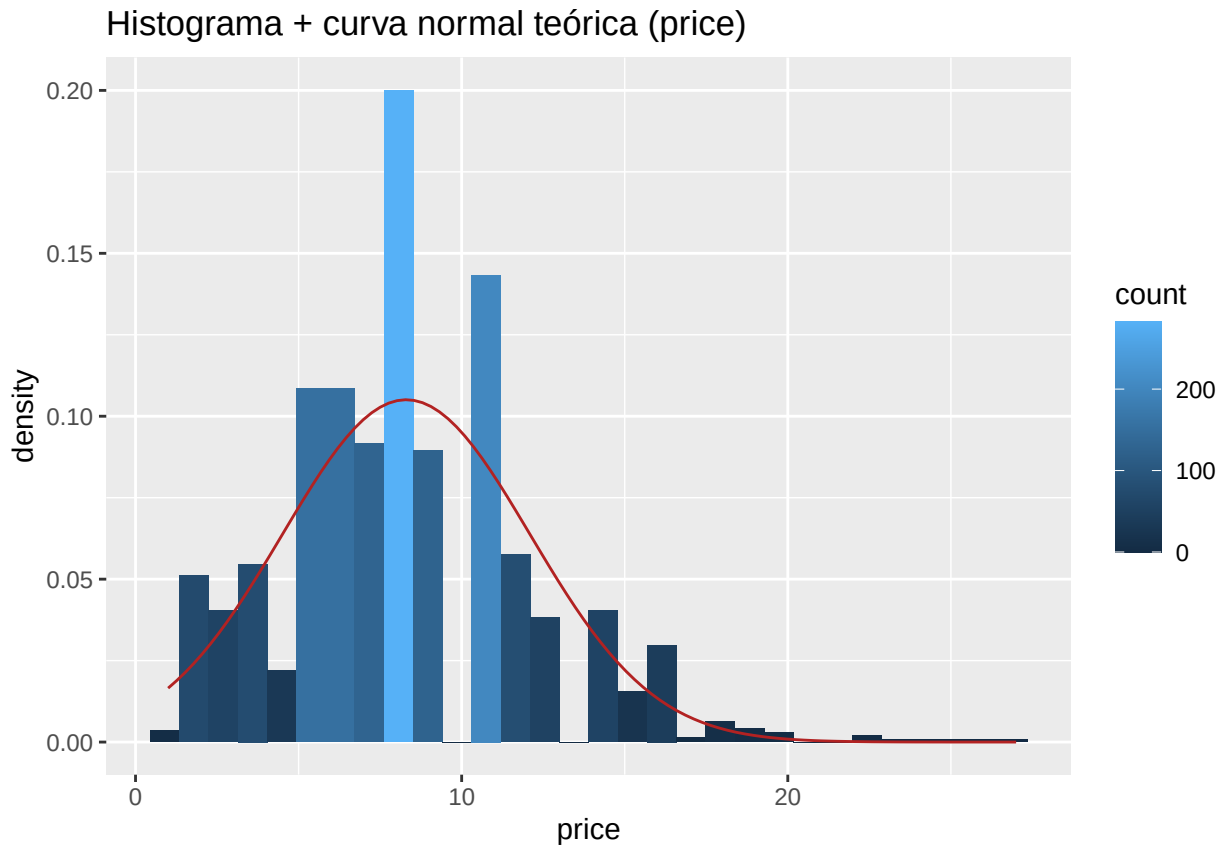
```
alpha = 0.05

col.names = colnames(SalesSummerObj)

for (i in 1:ncol(SalesSummerObj))
  {
    if (i == 1) cat("Variables que no siguen una distribución normal:\n")
      if (is.integer(SalesSummerObj[,i]) | is.numeric(SalesSummerObj[,i]))
        {
          p_val = shapiro.test(SalesSummerObj[,i])$p.value
          if (p_val < alpha) {
                                cat(col.names[i])
                                if (i < ncol(SalesSummerObj) - 1) cat(", ")
                                if (i %% 2 == 0) cat("\n")
                              }
        }
  }
```

```
## Variables que no siguen una distribución normal:
## price, retail_price,
## units_sold, uses_ad_boosts,
## rating, rating_count,
## rating_five_count, rating_four_count,
## rating_three_count, rating_two_count,
## rating_one_count, badges_count,
## badge_local_product, badge_product_quality,
## badge_fast_shipping, product_color, product_variation_inventory,
## shipping_is_express, countries_shipped_to,
## inventory_total, merchant_rating,
## rating_interval
```

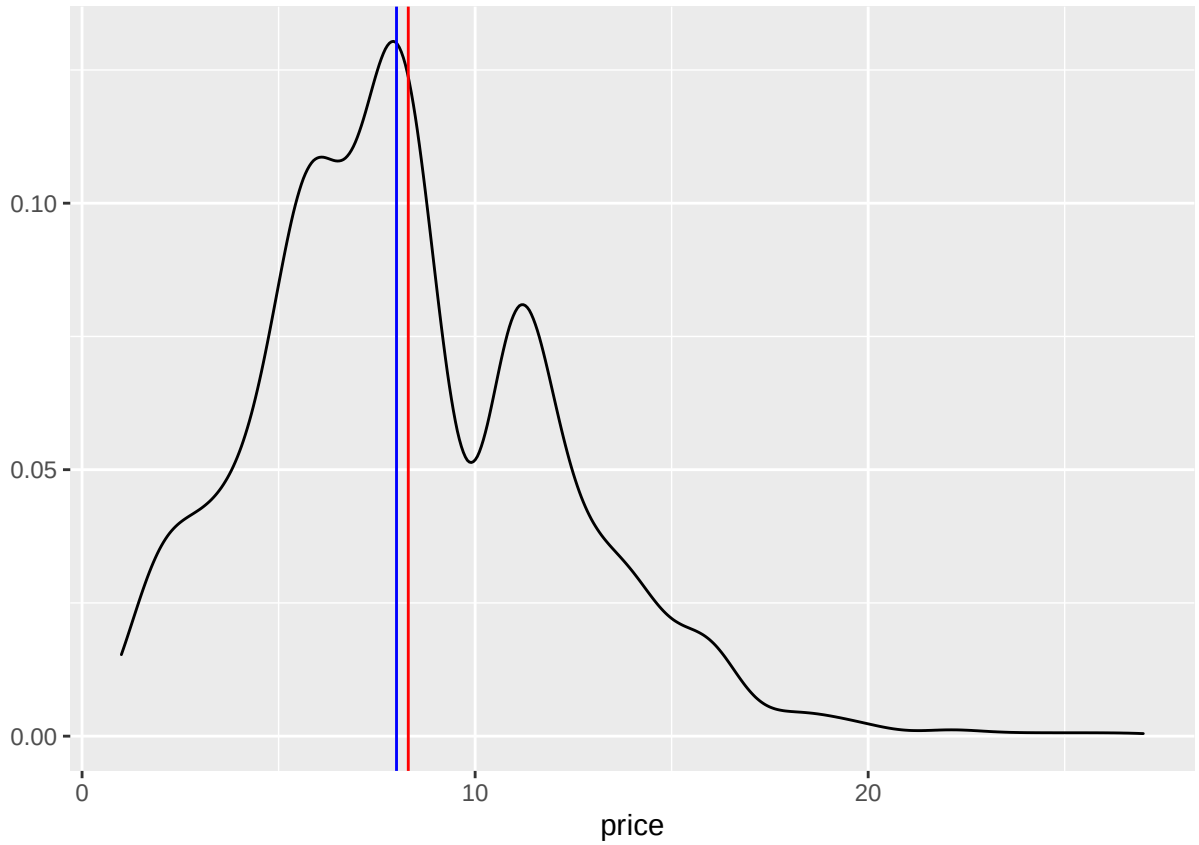
Podemos comprobarlo gráficamente por ejemplo con la variable price.

```
ggplot(data = SalesSummerObj, aes(x = price)) +
  geom_histogram(aes(y = ..density..,fill = ..count..)) +
  stat_function(fun = dnorm, colour = "firebrick",
                args = list(mean = mean(SalesSummerObj$price),
                            sd = sd(SalesSummerObj$price))) +
  ggtitle("Histograma + curva normal teórica (price)")
```

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




```
qplot(price, data = SalesSummerObj,geom="density")+ geom_vline(xintercept = mean(SalesSummerObj$price),
color="red")+ geom_vline(xintercept = median(SalesSummerObj$price), color="blue")
```



Ninguna de las variables seleccionadas es normal

Seguidamente, pasamos a estudiar la homogeneidad de varianzas.

Como ninguna de las variables es normal aplicaremos el test de Levene entre price y las variables que vamos a utilizar:

```
leveneTest(price ~ factor(SalesSummerObj$uses_ad_boosts), SalesSummerObj, center=median)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value   Pr(>F)
## group    1  10.415 0.001276 **
##       1570
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
leveneTest(price ~ factor(SalesSummerObj$badge_local_product), SalesSummerObj, center=median)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value  Pr(>F)
## group    1  5.9438 0.01488 *
##       1570
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
leveneTest(price ~ factor(SalesSummerObj$badge_product_quality), SalesSummerObj, center=median)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value Pr(>F)
## group    1  1.9869 0.1589
##       1570
```

```r
leveneTest(price ~ factor(SalesSummerObj$badge_fast_shipping), SalesSummerObj, center=median)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value   Pr(>F)
## group    1  10.727 0.001079 **
##       1570
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
leveneTest(price ~ factor(SalesSummerObj$shipping_is_express), SalesSummerObj, center=median)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value Pr(>F)
## group    1   0.096 0.7567
##       1570
```

```r
leveneTest(price ~ factor(SalesSummerObj$rating_interval), SalesSummerObj, center=median)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value Pr(>F)
## group    4  0.3153 0.8679
##       1567
```

```r
leveneTest(price ~ factor(SalesSummerObj$origin_country), SalesSummerObj, center=median)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value Pr(>F)
## group    6  1.0617 0.3835
##       1565
```

Los resultados indican que **no hay diferencias significativas** en los grupos de varianzas.(p-value > 0.05)

2.4.4 - Aplicación de pruebas estadísticas

2.4.4.1 - Estudio de la Correlación / Test de Spearman

En primer lugar, procedemos a realizar un análisis de correlación entre las distintas variables para determinar cuáles de ellas ejercen una mayor influencia sobre el precio del artículo. Para ello, se utilizará el coeficiente de correlación de Spearman, puesto que hemos visto que tenemos datos que no siguen una distribución normal.

```r
print(corr_matrix)
```

```
##                                estimate       p-value
## retail_price                0.535723785 1.776601e-117
## units_sold                  0.062816455  1.273645e-02
## uses_ad_boosts             -0.084007056  8.560815e-04
## rating                      0.053976502  3.235940e-02
## rating_count                0.131051999  1.844338e-07
## rating_five_count           0.136014159  6.175722e-08
## rating_four_count           0.136226187  5.888450e-08
## rating_three_count          0.121194525  1.441880e-06
## rating_two_count            0.113147200  6.892190e-06
## rating_one_count            0.124128744  7.945499e-07
## badges_count                0.019736163  4.342369e-01
## badge_local_product         0.053715310  3.320693e-02
## badge_product_quality       0.006966719  7.825455e-01
## badge_fast_shipping         0.013961882  5.801582e-01
## product_color              -0.038964919  1.225262e-01
## product_variation_inventory 0.330423088  2.333275e-41
## shipping_is_express         0.015170636  5.478077e-01
## countries_shipped_to       -0.016881306  5.036021e-01
## inventory_total            -0.051707039  4.037959e-02
## merchant_rating             0.054049615  3.212550e-02
## rating_interval             0.053574000  3.367331e-02
```

Los grados de correlación de las variables son tanto más altos, cuanto más cerca están de -1 o de 1. Teniendo esto en cuenta, la variable más relevante en la fijación del precio es el precio de retail (retail_price), seguida de product_variation_inventory, y las variables de rating_count. De cualquier forma, no hay ninguna variable que este correlacionada de forma fuerte con la variable price, ya que ninguna está por encima de 0.8

El valor P es la probabilidad de que hubiera encontrado el resultado actual si el coeficiente de correlación fuera cero (hipótesis nula). Si esta probabilidad es menor que el 5% convencional (P <0.05), el coeficiente de correlación se denomina estadísticamente significativo, por lo que modemos concluir que, el coeficiente de correlación entre retail_price y price (0.5358) es estadísticamente significativo, lo mismo sucede para las variables rating_*_count.

El valor del coeficiente positivo, indica una correlación positiva, es decir, cuanto mayor es el precio de retail, mayor es el precio del producto.

2.4.4.2 - Contraste de Hipótesis

Determinar si el precio es superior dependiendo del rating_interval del producto

Para ello utilizaremos dos muestras: una cuando la variable rating_interval del producto es cinco (rating >= 4.5) y otra cuando dicha variable es tres (rating <= 1.5)***

Para realizar este tipo de tests paramétricos, es preciso que los datos sean normales, si la muestra es de tamaño inferior a 30. En nuestro caso, el contraste de hipótesis es aplicable ya que superamos dicho valor.

Planteamos un contraste de Hipótesis unilateral sobre la diferencia de medias:

$$H_0 : \mu_1 - \mu_2 = 0$$

$$H_1 : \mu_1 - \mu_2 < 0$$

donde mu1 es la media de la población de la que se extrae la primera muestra y mu2 es la media de la población de la que extrae la segunda. Así, tomaremos alfa = 0.05.

```
t.test(SalesSummerObj.rating.interval.tres$price,SalesSummerObj.rating.interval.cinco$price,alternative
```

```
##
##  Welch Two Sample t-test
##
## data:  SalesSummerObj.rating.interval.tres$price and SalesSummerObj.rating.interval.cinco$price
## t = 0.46539, df = 197.86, p-value = 0.6789
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##       -Inf 0.8709991
## sample estimates:
## mean of x mean of y
##  7.888841  7.697455
```

Obtenemos un p-value mayor que el valor de significación, por lo que no podemos rechazar la hipotésis nula. Por lo tanto no podemos concluir que los artículos con un rating_interval = 5 sean más caros que los que tienen un rating_interval = 1

Determinar si el precio es superior para los productos de mayor calidad

Para ello utilizaremos dos muestras: una cuando la variable badget_product_quality del producto es 1 y otra cuando dicha variable es cero***

Aplicaremos el mismo test de hpótesis que en el caso anterior.

```
t.test(SalesSummerObj.badget.productquality.cero$price,SalesSummerObj.badget.productquality.uno$price,a
```

```
##
##  Welch Two Sample t-test
##
## data:  SalesSummerObj.badget.productquality.cero$price and SalesSummerObj.badget.productquality.uno$p
## t = -0.4366, df = 132.42, p-value = 0.3316
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##       -Inf 0.4762453
## sample estimates:
## mean of x mean of y
##  8.286811  8.457265
```

Obtenemos un p-value mayor que el valor de significación, por lo que no podemos rechazar la hipotésis nula. Por lo tanto no podemos concluir que los artículos catalogados de mayor calidad tengan un precio superior

Determinar si el uso de anuncios incrementa al precio del producto

Para ello utilizaremos dos muestras: una cuando la variable uses_ad_boosts del producto es 1 y otra cuando dicha variable es cero***

Aplicaremos el mismo test de hpótesis que en el caso anterior.

```
t.test(SalesSummerObj.uses.ad.boosts.cero$price,SalesSummerObj.uses.ad.boosts.uno$price,alternative = "
```

```
##
##  Welch Two Sample t-test
##
## data:  SalesSummerObj.uses.ad.boosts.cero$price and SalesSummerObj.uses.ad.boosts.uno$price
## t = 2.725, df = 1342.4, p-value = 0.9967
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##       -Inf 0.8595862
## sample estimates:
## mean of x mean of y
##  8.531650  7.995756
```

Obtenemos un p-value mayor que el valor de significación, por lo que no podemos rechazar la hipotésis nula. Por lo tanto no podemos concluir que el uso de anuncios incrementa el precio de venta.

Determinar si el envío rápido incrementa al precio del producto

NO podemos aplicar el test para este grupo de datos ya que la muestra para fastshipping.uno es $19 < 30$ y la variable no es normal.

Determinar si el envío express incrementa al precio del producto

NO podemos aplicar el test para este grupo de datos ya que la muestra para shipping.uno es $3 < 30$ y la variable no es normal.

2.4.4.3 - Regresión lineal

Plantearemos varios modelos de regresión utilizando algunos de los regresores cuantitativos que tengan la correlación más alta con la variable precio:

retail_price, rating_count, rating_five_count, rating_four_count, rating_three_count, rating_two_count, rating_one_count, product_variation_inventory

```
# regresores
retprice = SalesSummerObj$retail_price
pvinventory = SalesSummerObj$product_variation_inventory
ratingone = SalesSummerObj$rating_one_count
ratingfour = SalesSummerObj$rating_four_count

# variable independiente

artprice = SalesSummerObj$price
```

Modelo1 : Utilizando los regresores retprice y pvinventory.

```
# modelo1


modelo1 <- lm(artprice ~ retprice + pvinventory, data = SalesSummerObj)

summary(modelo1)
```

```
##
## Call:
```

```
## lm(formula = artprice ~ retprice + pvinventory, data = SalesSummerObj)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.4284 -2.4795 -0.5145  2.3098 17.9247
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 5.944997   0.169618   35.05   <2e-16 ***
## retprice    0.035050   0.002907   12.06   <2e-16 ***
## pvinventory 0.046483   0.004135   11.24   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.473 on 1569 degrees of freedom
## Multiple R-squared:  0.1644, Adjusted R-squared:  0.1633
## F-statistic: 154.3 on 2 and 1569 DF,  p-value: < 2.2e-16
```

Modelo2: Añadiendo el modelo anterior los regresores ratingfour y rating one

```
modelo2 <- lm(artprice ~ retprice + pvinventory + ratingfour + ratingone, data = SalesSummerObj)

summary(modelo2)
```

```
##
## Call:
## lm(formula = artprice ~ retprice + pvinventory + ratingfour +
##     ratingone, data = SalesSummerObj)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.5412 -2.5173 -0.5077  2.2658 17.8279
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.9817494  0.1705644  35.070   <2e-16 ***
## retprice     0.0353308  0.0029077  12.151   <2e-16 ***
## pvinventory  0.0475773  0.0041680  11.415   <2e-16 ***
## ratingfour  -0.0005485  0.0004909  -1.117    0.264
## ratingone    0.0001746  0.0009177   0.190    0.849
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.471 on 1567 degrees of freedom
## Multiple R-squared:  0.1667, Adjusted R-squared:  0.1646
## F-statistic: 78.36 on 4 and 1567 DF,  p-value: < 2.2e-16
```

Calidad del ajuste de los modelos

El Coeficiente de determinación R2 mide el grado en el que el modelo de regresión lineal explica la variaciones que se producen en la variable dependiente de las observaciones, y se calcula dividiendo la varianza explicada por la recta de regresión entre la varianza total de los datos

$$R^2 = \frac{\sigma rectaregresión}{\sigma totaldatos} = \frac{SCR}{SCT} = 1 - \frac{SCE}{SCT}$$

Este coeficiente aparecía al calcular ambos modelos con la función lm(), en nuestro caso es 0.1644 en el modelo1, pero conviene utilizar el ajustado que es: **0.1633**

Lo que implica que el modelo es capaz de explicar alrededor del **16 % de la varianza de las observaciones**. Resultado bastante pobre.

El coeficiente de correlación muestral, r, mide el grado de asociación entre las variables. Vamos a calcularlo a partir del coeficiente de determinación:

```
r1 = sqrt(0.1633)
```

Lo que indica una relación lineal no excesivamente fuerte entre las variables utilizadas y el precio.

A continuación vamos a calcular los intervalos de confianza del modelo1:

```
confint(modelo1)
```

```
##                   2.5 %     97.5 %
## (Intercept) 5.61229552 6.27769791
## retprice    0.02934881 0.04075172
## pvinventory 0.03837288 0.05459343
```

Vemos que el intervalo menos amplio corresponde al regresor retprice, por lo que indica mayor precisión. Es decir, mientras más confianza se necesita, más ancho es el intervalo.

Observamos que la mayor ineficiencia en la estimación del parámetro corresponde al regresor pvinventory, pero con una diferencia realmene mínima respecto a retprice

En el segundo modelo comprobamos que el hecho de añadir los regresores ratingfour y ratingone no han hecho variar el coeficiente de determinación ajustado, que se queda en un 16%, por lo que se descarta la utilización de estos regresores. Es decir, estos regresores no aportan nada en cuanto a la capacidad de predicción del modelo.

Realizaremos una predicción del precio de venta:

```
newdata <- data.frame(retprice=30,pvinventory=50)

# Predecir el precio

predict(modelo1, newdata)
```

```
##        1
## 9.320662
```

2.4.4.4 - Regresión Logística (Multinomial)

Se trata de un modelo de regresión logística donde la variable dependiente tiene más de dos categorías . La respuesta puede o bien ser nominal o bien ordinal. A su vez, las variables explicativas pueden ser categóricas o cuantitativas.

En este caso vamos a tratar como variable dependiente la variable (size_category (EC,HS,SS)), y como variable independiente origin_country (AT,CN,GB,NC,US,VE).

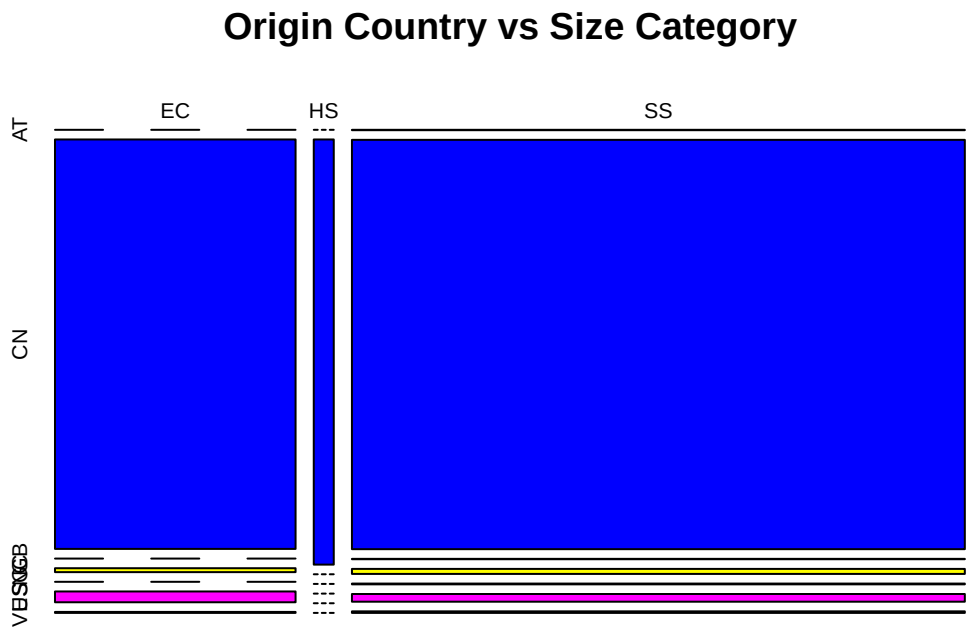
Las variables que vamos a utilizar en este modelo son cuantitativas, por lo que el análisis de sus relaciones se ha de obtener mediante Tablas de contingencia y pruebas Chi-Cuadrado

2.4.4.4.1 Tablas de Contingencia

```
Tabla.SC.OC <- table(SalesSummerObj$size_category,SalesSummerObj$origin_country)

Tabla.SC.OC
```

```
##
##         AT   CN   GB   NC   SG   US   VE
##    EC    0  417    0    4    0   11    1
##    HS    0   36    0    0    0    0    0
##    SS    1 1062    1   13    2   20    4
```

```
plot(Tabla.SC.OC, col= c("red", "blue", "green", "yellow", "cyan", "magenta", "orange"),
     main = "Origin Country vs Size Category")
```



Vemos que la mayor contribución a los tipos de tallas viene de China, especialmente sobre las tallas pequeñas. El siguiente país es Estados Unidos, sobre todo en tallas pequeñas también. Hay una contribución de ventas no catalogadas por país en las tallas grandes. Y países como China que contribuyen en gran medida a ventas donde no se han indicado las tallas.

2.4.4.4.2 Estudio de la Correlación / Tests Chi-Squared

Estamos tratando variables cuantitativas politómicas y nominales, por lo que, para valorar la independencia, el test Chi-squared resulta adecuado en algunos casos, y el test exacto de Fisher en otros.

```r
chisq.test(Tabla.SC.OC)
```

```
## Warning in chisq.test(Tabla.SC.OC): Chi-squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  Tabla.SC.OC
## X-squared = 4.1833, df = 12, p-value = 0.9799
```

```r
fisher.test(Tabla.SC.OC, conf.level = 0.95, simulate.p.value = FALSE)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  Tabla.SC.OC
## p-value = 0.9762
## alternative hypothesis: two.sided
```

Como el **p-value es $>$ 0.05** no podemos rechazar la hipotésis nula, que indica independencia entre ambas variables. Por lo tanto no existe correlación entre ellas.

Cálculo del modelo

```r
model.sizecategory.origincountry = multinom(SalesSummerObj$origin_country ~
SalesSummerObj$size_category, data = SalesSummerObj)
```

```r
# Obtenemos el summary

summary(model.sizecategory.origincountry)
```

```
## Call:
## multinom(formula = SalesSummerObj$origin_country ~ SalesSummerObj$size_category,
##     data = SalesSummerObj)
##
## Coefficients:
##    (Intercept) SalesSummerObj$size_categoryHS SalesSummerObj$size_categorySS
## CN    17.26118                      16.353274                     -10.293432
## GB   -15.91788                      -2.443693                      15.917608
## NC    12.61436                      -2.835709                     -10.049563
## SG   -15.57150                      -2.443693                      16.264657
## US    13.62595                      -3.737094                     -10.630380
## VE    11.22853                      -2.449248                      -9.842395
##
## Std. Errors:
##    (Intercept) SalesSummerObj$size_categoryHS SalesSummerObj$size_categorySS
## CN   0.5839464                   6.426676e-10                      0.5845474
## GB   0.7070966                   1.536329e-15                      0.7070966
## NC   0.6817606                   3.790453e-10                      0.7228307
## SG   0.6123201                   5.336587e-22                      0.6123201
## US   0.6206858                   1.208965e-10                      0.6501897
```

```
## VE   0.9162833                    9.293063e-10                         1.0134429
##
## Residual Deviance: 616.3715
## AIC: 652.3715
```

```
# Coefcientes Modelo

coefmodel.sizecategory.origincountry <- coef(model.sizecategory.origincountry)

coefmodel.sizecategory.origincountry
```

```
##    (Intercept) SalesSummerObj$size_categoryHS SalesSummerObj$size_categorySS
## CN    17.26118                      16.353274                      -10.293432
## GB   -15.91788                      -2.443693                       15.917608
## NC    12.61436                      -2.835709                      -10.049563
## SG   -15.57150                      -2.443693                       16.264657
## US    13.62595                      -3.737094                      -10.630380
## VE    11.22853                      -2.449248                       -9.842395
```

Vamos a evaluar ahora los ***odds ratio***. Los ***odds*** es la razón de la probabilidad de ocurrencia de un suceso entre la probabilidad de su no ocurrencia.Vamos a ver como transformamos los coeficientes en odds ratios. Trataremos de ser algo didácticos ,y vamos a explicar en detalle su cálculo para China:

En esta expresión, el modelo está expresado en términos del ***log-odds***:

$$ln(\frac{P(Y=1/X)}{1-P(Y=1/X)}) = 17.261 + 16.353 * HS - 10.293 * SS$$

Si se escribe en términos de odds, se tiene:

$$\frac{P(Y=1/X)}{1-P(Y=1/X)} = \frac{e^{b_0} + \sum_{i=1}^{n}(b_i x_i)}{1 + e^{b_0} + \sum_{i=1}^{n}(b_i x_i)}$$

Se calculan los distintos valores de las probabilidades para las cuatro combinaciones entre la variable dependiente Y con la independiente X:

$$\frac{P(Y=1/X=1)}{1-P(Y=1/X=1)} = \frac{e^{b_0+b_1}}{1+e^{b_0+b_1}}$$

$$\frac{P(Y=1/X=0)}{1-P(Y=1/X=0)} = \frac{e^{b_0}}{1+e^{b_0}}$$

$$\frac{P(Y=0/X=1)}{1-P(Y=0/X=1)} = \frac{1}{1+e^{b_0+b_1}}$$

$$\frac{P(Y=0/X=0)}{1-P(Y=0/X=0)} = \frac{1}{1+e^{b_0}}$$

Los ***odds-ratio (OR)*** se calculan como la razón entre los ***odds***, donde la variable respuesta Y está presente entre los individuos, es decir, toma el valor Y = 1, y la variable independiente X puede estar presente o no, es decir, tomar los valores X = 1 y X = 0.

$$OR = \frac{\frac{P(Y=1/X=1)}{1-P(Y=1/X=1)}}{\frac{P(Y=1/X=0)}{1-P(Y=1/X=0)}} = e^{b_1}$$

• Un OR = 1 implica que no existe asociación entre la variable respuesta y la covariable. • Un OR inferior a la unidad se interpreta como un factor de protección, es decir, el suceso es menos probable en presencia de dicha covariable. • Un OR mayor a la unidad se interpreta como un factor de riesgo, es decir,el suceso es más probable en presencia de dicha covariable.

Para el caso de los Estados Unidos:

$$ln(\frac{P(Y=1/X)}{1-P(Y=1/X)}) = 13.626 - 3.737 * HS - 10.630 * SS$$

```
# Odds Ratios Modelo


exp(coefmodel.sizecategory.origincountry)
```

```
##     (Intercept) SalesSummerObj$size_categoryHS SalesSummerObj$size_categorySS
## CN 3.136418e+07                   1.265134e+07                   3.385472e-05
## GB 1.221665e-07                   8.683955e-02                   8.183316e+06
## NC 3.008489e+05                   5.867692e-02                   4.320462e-05
## SG 1.727371e-07                   8.683955e-02                   1.157846e+07
## US 8.273211e+05                   2.382323e-02                   2.417043e-05
## VE 7.524687e+04                   8.635854e-02                   5.314987e-05
```

Calcularemos ahora los intervalos de confianza:

```
# Intervalos de confianza odds ratio


Modelo.sizecategory.origincountry <- confint(model.sizecategory.origincountry)

Modelo.sizecategory.origincountry
```

```
## , , CN
##
##                                    2.5 %   97.5 %
## (Intercept)                     16.11666 18.40569
## SalesSummerObj$size_categoryHS  16.35327 16.35327
## SalesSummerObj$size_categorySS -11.43912 -9.14774
##
## , , GB
##
##                                     2.5 %     97.5 %
## (Intercept)                    -17.303765 -14.531997
## SalesSummerObj$size_categoryHS  -2.443693  -2.443693
## SalesSummerObj$size_categorySS  14.531724  17.303492
##
## , , NC
##
```

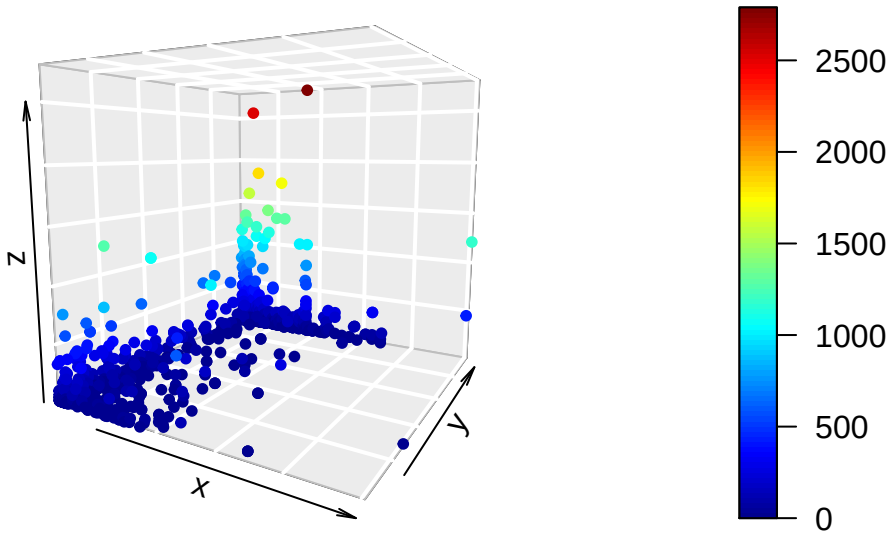
```
##                                      2.5 %    97.5 %
## (Intercept)                       11.278137 13.950590
## SalesSummerObj$size_categoryHS    -2.835709 -2.835709
## SalesSummerObj$size_categorySS   -11.466285 -8.632841
##
## , , SG
##
##                                      2.5 %     97.5 %
## (Intercept)                      -16.771621 -14.371370
## SalesSummerObj$size_categoryHS    -2.443693  -2.443693
## SalesSummerObj$size_categorySS    15.064532  17.464783
##
## , , US
##
##                                      2.5 %    97.5 %
## (Intercept)                       12.409426 14.842470
## SalesSummerObj$size_categoryHS    -3.737094 -3.737094
## SalesSummerObj$size_categorySS   -11.904729 -9.356032
##
## , , VE
##
##                                      2.5 %    97.5 %
## (Intercept)                        9.432647 13.024412
## SalesSummerObj$size_categoryHS    -2.449248 -2.449248
## SalesSummerObj$size_categorySS   -11.828706 -7.856083
```

2.5 - REPRESENTACIÓN DE RESULTADOS

Interpretación de Modelos

Vamos a representar gráficamente los datos de los regresores retprice, pvinventory y ratingone respecto a artprice:

```
scatter3D(x=retprice, y=pvinventory, z =ratingone ,groups=artprice, theta=30, phi=8, pch=20, bty = "g",
          grid=FALSE, fit="smooth")
```



Vemos que los precios más bajos se concentran para valores bajos y altos de inventario de la talla de la compra, y para precios de retail bajos. Los precios medios se agrupan en los valores centrales de los tres regresores, y los precios más altos se encuentran para valores centrales de los regresores retprice y pvinventory, y para valores altos del regresor ratingone.

Tabla resumen modelo regresión logística multinomial

2.6 - RESOLUCIÓN DEL PROBLEMA

REFERENCIAS

Documentación máster UOC: Modelos_de_Regresión_Logística.pdf (PID_00276229)

6 Errores que cometes al usar las pruebas de hipótesis clásicas: https://www.maximaformacion.es/blog-dat/6-errores-que-cometes-al-usar-las-pruebas-de-hipotesis-clasicas/

Modelos con Variables Cualitativas: https://bookdown.org/content/2274/modelos-con-variables-cualitativas.html

Estadística conceptos clave: https://www.usj.es/sites/default/files/tarjetas/aprendizaje/EstadisticaConceptosClave.pdf

Análisis de variables categóricas con R: https://biocosas.github.io/R/060_analisis_datos_categoricos.html

Correlación: teoría y práctica: https://www.ccg.unam.mx/~vinuesa/R4biosciences/docs/Tema8_correlacion.html

Test estadísticos para variables cualitativas: test exacto de Fisher, chi-cuadrado de Pearson, Mc-Nemar y Q-Cochran: https://www.cienciadedatos.net/documentos/22.2_test_exacto_de_fisher_chi-cuadrado_de_pearson_mcnemar_qcochran#(chi%5E2)_de_Pearson_(test_de_independencia)

Logistic Regression in R: https://rpubs.com/rslbliss/r_logistic_ws

Multinomial distribution: https://en.wikipedia.org/wiki/Multinomial_distribution

Test estadísticos para variables cualitativas: test binomial exacto, test multinomial y test chi-cuadrado goodnes of fit:https://www.cienciadedatos.net/documentos/22.1_test_binomial_exacto_test_multinomial_test_chi-cuadrado_goodnes_of_fit

Modelos de respuesta multinomial con R. Aplicación para el estudio de la depresión en pacientes con discapacidad: https://masteres.ugr.es/moea/pages/tfm1011/modelosderespuestamultinomialconraplicacionparaelestudiodeladep

Regresión Logística Multinomial: Multinomialhttp://halweb.uc3m.es/esp/Personal/personas/jmmarin/esp/Categor/Tema5C (Modelos Logit para respuestas nominales)