

Limpieza y Análisis de Datos

Diciembre 2020

Contents

1 - DESCRIPCIÓN ACTIVIDAD	2
1.1 - OBJETIVOS	2
1.2 - COMPETENCIAS	2
2 - RESOLUCIÓN	3
2.1 - DESCRIPCIÓN DEL DATASET / IMPORTANCIA	3
2.2 - INTEGRACIÓN Y SELECCIÓN DE DATOS	6
2.3 - LIMPIEZA DE LOS DATOS	6
2.3.1 - Selección de los datos de interes	7
2.3.2 - Ceros y elementos vacíos	7
2.3.3 - Identificación y tratamiento de outliers	13
2.3.4 - Exportación de los datos preprocesados	17
2.3.5 - Factorización y niveles de las variables cuantitativas	17
2.4 - ANÁLISIS DE LOS DATOS	18
2.4.2 - Selección de grupos de datos	18
2.4.3 - Comprobación de normalidad y homogeneidad de la varianza	19
2.4.4 - Aplicación de pruebas estadísticas	23
2.4.4.1 - Estudio de la Correlación / Test de Spearman	23
2.4.4.2 - Contraste de Hipótesis	24
2.4.4.3 - Regresión lineal	26
2.4.4.4 - Regresión Logística (Multinomial)	28
2.4.4.4.1 Tablas de Contingencia	29
2.4.4.4.2 Estudio de la Correlación / Tests Chi-Squared	29

2.5 - REPRESENTACIÓN DE RESULTADOS	33
2.6 - RESOLUCIÓN DEL PROBLEMA	34
REFERENCIAS	34

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/jmmvutu/summer-products-and-sales-in-ecommerce-wish>, y contiene información sobre las ventas de productos de Verano en la plataforma ecommerce **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 qué forma, sobre el precio del producto. También plantearemos algunas pruebas de contrastes de hipótesis, para confirmar o desmentir hechos que planteemos una vez analizados los datos y modelos de regresión para ver cómo se 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 qué 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 también de determinar qué 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, inventory_total, merchant_rating, product_variation_size_id, origin_country

2.3 - LIMPIEZA DE LOS DATOS

Se realiza una inspección preliminar 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 proporcionados 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 dichos 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 vacíos

colSums(is.na(SalesSummerObj) | SalesSummerObj=="")
```

```
##          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
##              41              0
##      shipping_is_express      countries_shipped_to
##              0              0
##      inventory_total      merchant_rating
##              0              0
##      product_variation_size_id      origin_country
##              14              17
```

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:

$$\text{rating} = \text{rating5} * 5 + \text{rating4} * 4 + \text{rating3} * 3 + \text{rating2} * 2 + \text{rating1} / \text{rating_count}$$

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$product_color <- as.character(SalesSummerObj$product_color)
SalesSummerObj$product_color[SalesSummerObj$product_color==""] <- "no color"
SalesSummerObj$product_color <- factor(SalesSummerObj$product_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"] <- "army green"
SalesSummerObj$product_color[SalesSummerObj$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"
SalesSummerObj$product_color[SalesSummerObj$product_color=="Black"] <- "black"
SalesSummerObj$product_color[SalesSummerObj$product_color=="blackwhite"] <- "black & white"

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

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

```
SalesSummerObj$origin_country <- as.character(SalesSummerObj$origin_country)
SalesSummerObj$origin_country[SalesSummerObj$origin_country==""] <- "NC"
SalesSummerObj$origin_country <- factor(SalesSummerObj$origin_country)
```

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

```
SalesSummerObj$product_variation_size_id <- as.character(SalesSummerObj$product_variation_size_id)
SalesSummerObj$product_variation_size_id[SalesSummerObj$product_variation_size_id == ""] <- "No size"
SalesSummerObj$product_variation_size_id <- factor(SalesSummerObj$product_variation_size_id)
```

La variable `product_variation_size_id` tiene diferentes valores que hacen referencia a una misma talla. Unificamos estos valores:

```
SalesSummerObj$product_variation_size_id <- as.character(SalesSummerObj$product_variation_size_id)

# Talla 3XS
SalesSummerObj$product_variation_size_id[SalesSummerObj$product_variation_size_id=="XXXS"] <- "3XS"
```



```

# Talla 3XL
SalesSummerObj$product_variation_size_id[SalesSummerObj$product_variation_size_id=="XXXL"] <- "3XL"
SalesSummerObj$product_variation_size_id[SalesSummerObj$product_variation_size_id=="04-3XL"] <- "3XL"

# Talla 4XL
SalesSummerObj$product_variation_size_id[SalesSummerObj$product_variation_size_id=="SIZE-4XL"] <- "4XL"
SalesSummerObj$product_variation_size_id[SalesSummerObj$product_variation_size_id=="Size4XL"] <- "4XL"
SalesSummerObj$product_variation_size_id[SalesSummerObj$product_variation_size_id=="XXXXL"] <- "4XL"

# Talla 5XL
SalesSummerObj$product_variation_size_id[SalesSummerObj$product_variation_size_id=="Size-5XL"] <- "5XL"
SalesSummerObj$product_variation_size_id[SalesSummerObj$product_variation_size_id=="XXXXXL"] <- "5XL"

# Sin talla
SalesSummerObj$product_variation_size_id[SalesSummerObj$product_variation_size_id == "choose a size"] <- "No size"
SalesSummerObj$product_variation_size_id[SalesSummerObj$product_variation_size_id == "Pack of 1"] <- "No size"
SalesSummerObj$product_variation_size_id[SalesSummerObj$product_variation_size_id == "5PAIRS"] <- "No size"
SalesSummerObj$product_variation_size_id[SalesSummerObj$product_variation_size_id == "Round"] <- "No size"
SalesSummerObj$product_variation_size_id[SalesSummerObj$product_variation_size_id == "White"] <- "No size"
SalesSummerObj$product_variation_size_id[SalesSummerObj$product_variation_size_id == "Base & Top & Matte"] <- "No size"
SalesSummerObj$product_variation_size_id[SalesSummerObj$product_variation_size_id == "Base Coat"] <- "No size"
SalesSummerObj$product_variation_size_id[SalesSummerObj$product_variation_size_id == "AU plug Low quality"] <- "No size"
SalesSummerObj$product_variation_size_id[SalesSummerObj$product_variation_size_id == "B"] <- "No size"

SalesSummerObj$product_variation_size_id <- factor(SalesSummerObj$product_variation_size_id)

levels(SalesSummerObj$product_variation_size_id)

```

```

## [1] "1" "1 PC - XL"
## [3] "1 pc." "10 ml"
## [5] "100 cm" "100 x 100cm(39.3 x 39.3inch)"
## [7] "100pcs" "10pcs"
## [9] "17" "1m by 3m"
## [11] "1pc" "2"
## [13] "20pcs" "20PCS-10PAIRS"
## [15] "25" "26(Waist 72cm 28inch)"
## [17] "29" "2pcs"
## [19] "2XL" "2XS"
## [21] "3 layered anklet" "30 cm"
## [23] "33" "34"
## [25] "35" "36"
## [27] "3XL" "3XS"
## [29] "4" "4-5 Years"
## [31] "40 cm" "4XL"
## [33] "5" "5XL"
## [35] "60" "6XL"
## [37] "80 X 200 CM" "Baby Float Boat"
## [39] "daughter 24M" "EU 35"
## [41] "EU39(US8)" "first generation"
## [43] "Floating Chair for Kid" "H01"
## [45] "L" "M"
## [47] "No size" "One Size"

```

```
## [49] "S" "US 6.5 (EU 37)"
## [51] "US5.5-EU35" "Women Size 36"
## [53] "Women Size 37" "XL"
## [55] "XS"
```

size_category (EC,HS,SS)

```
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   black :305   Min.   : 1.00
## Class :character white :257   1st Qu.: 6.00
## Mode :character yellow:105   Median :50.00
##      pink :101   Mean   :33.08
##      blue : 99   3rd Qu.:50.00
##      red  : 94   Max.   :50.00
##      (Other):612
## 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
## S :693 AT: 1
## XS :369 CN:1516
## M :206 GB: 1
## 2XS :107 NC: 17
## L :55 SG: 2
## No size: 23 US: 31
## (Other):120 VE: 5
```

2.3.3 - Identificación y tratamiento de outliers

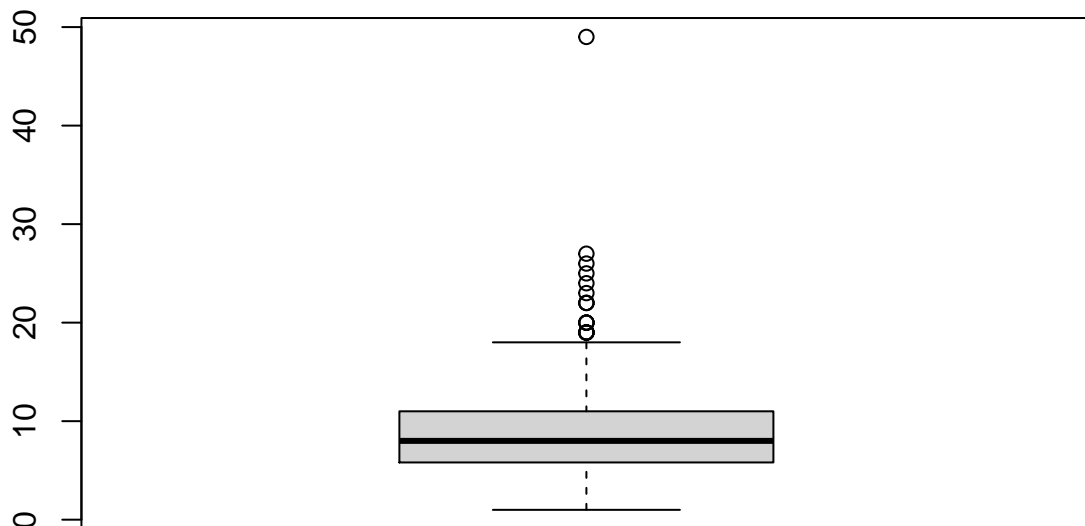
Un outlier es una observación anormal y extrema en una muestra estadística o serie temporal de datos, que puede afectar potencialmente a la estimación de los parámetros del mismo.

```
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      black :305   Min.    : 1.00
## Class :character  white  :257   1st Qu.: 6.00
## Mode  :character  yellow:105   Median :50.00
##                pink   :101   Mean   :33.08
##                blue   : 99   3rd Qu.:50.00
##                red    : 94   Max.    :50.00
##                (Other):612
## 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
## S      :693              AT: 1
## XS     :369              CN:1516
## M      :206              GB: 1
## 2XS    :107              NC: 17
## L      : 55              SG: 2
## No size: 23              US: 31
## (Other):120              VE: 5
```

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



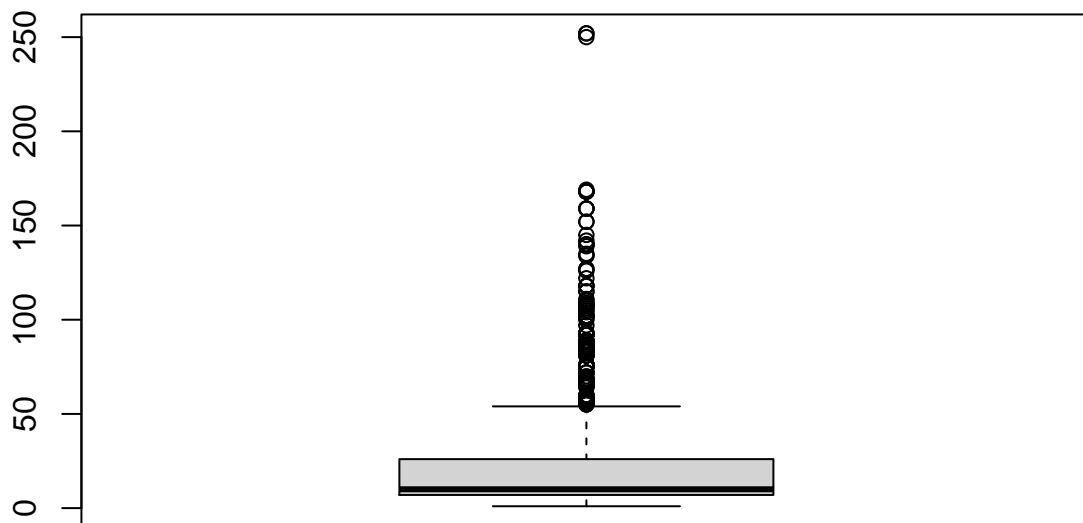
```
boxplot.stats(SalesSummerObj$price)$out
```

```
## [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 que lo contiene del conjunto de datos.

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

```
boxplot(SalesSummerObj$retail_price)
```

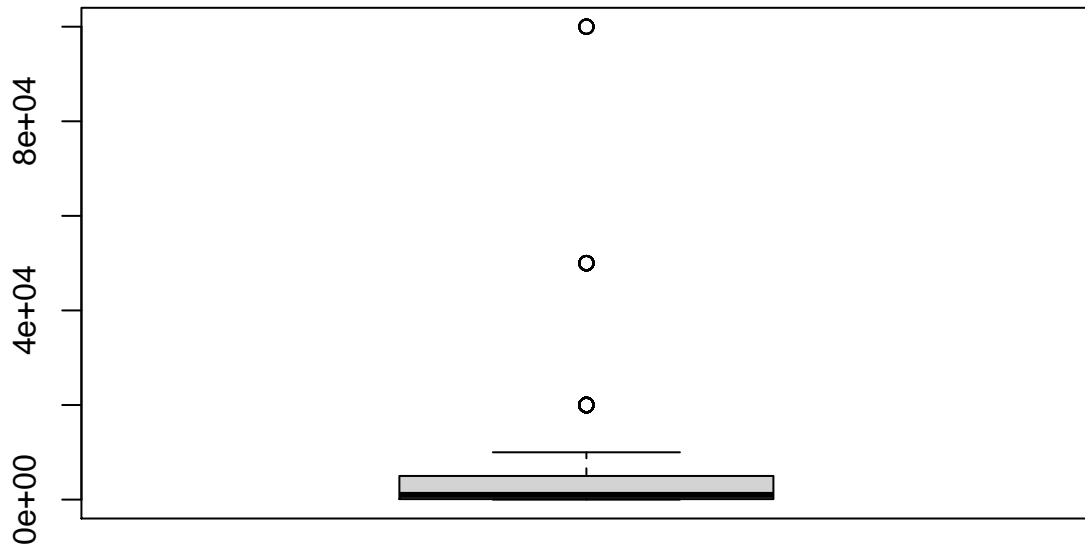


```
boxplot.stats(SalesSummerObj$retail_price)$out
```

```
## [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
```

En la variable `retail_price` existen dos valores extremos, pero al tratarse de precios podemos considerarlos valores válidos así que no los consideraremos outliers y no los eliminaremos del conjunto de datos.

```
boxplot(SalesSummerObj$units_sold)
```



El mismo caso se repite con las unidades vendidas, pero estaríamos hablando de una gran cantidad de unidades vendidas por encima de la media del conjunto de datos. Estos valores pueden provocar que los resultados en nuestro estudio se vean afectados. Así que decidimos eliminar dichos registros.

```
SalesSummerObj <- SalesSummerObj[!(SalesSummerObj$units_sold > 2000),]
```

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

```
## [1] 1.50 2.00 2.00 2.00 2.00 1.00 2.00 2.25 2.00 2.00 1.00 2.33 2.00 2.44 2.00
## [16] 1.50 2.33 1.00 2.44
```

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. 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 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] 3.298507 3.186047 3.473684 3.409471 3.034483 5.000000 2.941176 3.417722
## [9] 3.409471 3.381868 3.038961 3.475584 2.333333 3.464286 3.338290 3.381868
## [17] 4.577519 3.187500 3.186047 3.187500 3.367133 3.250000 3.422535 3.475584
## [25] 3.000000
```

Se trata de un rating que va entre 2.333 y 5, no vamos a categorizarlos 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 & blue"     "black & green"    "black & white"
## [10] "black & yellow"   "blue"             "brown"
## [13] "brown & yellow"   "camel"            "camouflage"
## [16] "claret"          "coffee"          "coolblack"
## [19] "coralred"        "darkblue"         "darkgreen"
## [22] "dustypink"       "floral"           "fluorescentgreen"
## [25] "gray"            "gray & white"     "green"
## [28] "grey"            "greysnakeskinprint" "khaki"
## [31] "lakeblue"        "leopard"          "leopardprint"
## [34] "lightblue"       "lightgray"        "lightgreen"
## [37] "lightgrey"       "lightpink"        "lightpurple"
## [40] "lightred"        "lightyellow"      "mintgreen"
## [43] "multicolor"      "navy"             "navyblue"
## [46] "no color"        "offblack"         "offwhite"
## [49] "orange"          "orange-red"       "orange & camouflage"
## [52] "pink"            "pink & black"     "pink & blue"
```

```
## [55] "pink & grey"      "pink & white"      "prussianblue"
## [58] "purple"           "rainbow"           "red"
## [61] "red & blue"        "rose"              "rosered"
## [64] "silver"           "skyblue"           "tan"
## [67] "violet"           "white"             "white & black"
## [70] "white & green"     "whitefloral"       "wine"
## [73] "winered"          "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 == "3XS" |
SalesSummerObj$product_variation_size_id == "2XS" | SalesSummerObj$product_variation_size_id == "XS" |
SalesSummerObj$product_variation_size_id == "S"] <- "SS"

SalesSummerObj$size_category[SalesSummerObj$product_variation_size_id == "XL" |
SalesSummerObj$product_variation_size_id == "2XL" | SalesSummerObj$product_variation_size_id == "3XL" |
SalesSummerObj$product_variation_size_id == "4XL" | SalesSummerObj$product_variation_size_id == "5XL" |
SalesSummerObj$product_variation_size_id == "6XL"] <- "HS"

```

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. Se comprueba si el p-valor es superior al nivel de significación prefijado $\alpha = 0.05$. 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)")

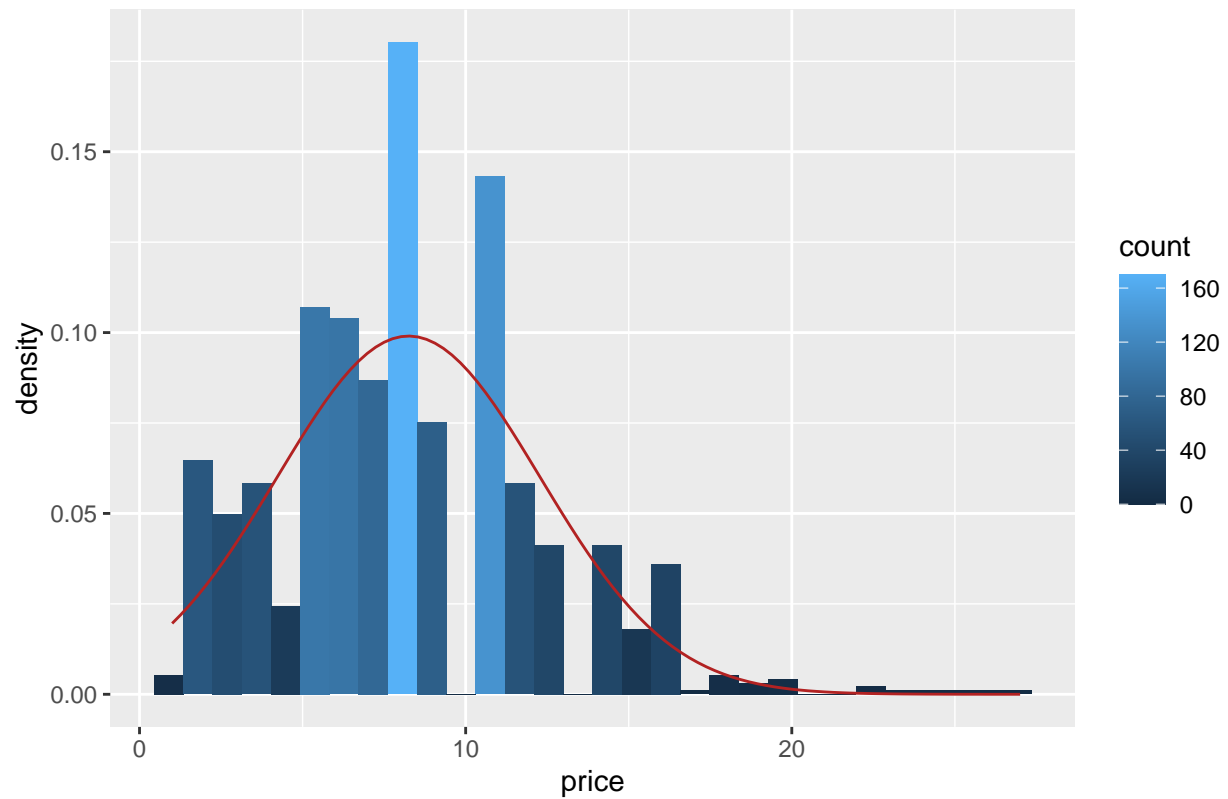
```

```

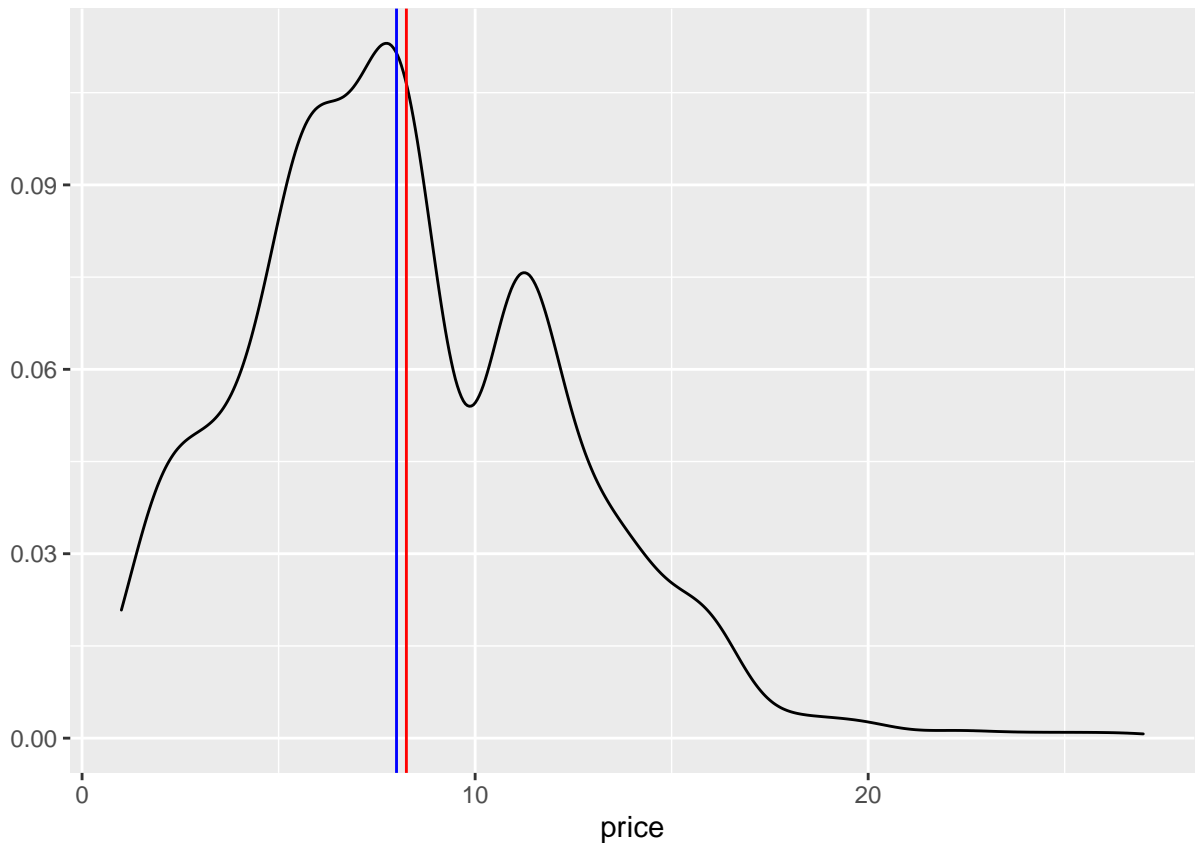
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

```

Histograma + curva normal teórica (price)



```
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. Este test prueba la hipótesis nula de que las varianzas poblacionales son iguales.

```
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   11.32 0.0007944 ***
##      1050
## ---
## 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    0.91 0.3403
##      1050
```

```
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  2.0329 0.1542
##           1050
```

```
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 12.118 0.0005198 ***
##           1050
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
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  0.2407 0.6238
##           1050
```

```
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  0.5301 0.7136
##           1047
```

```
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  1.4923 0.1774
##           1045
```

Si el P-valor resultante de la prueba de Levene es inferior a un cierto nivel de significación (típicamente 0.05), es poco probable que las diferencias obtenidas en las variaciones de la muestra se hayan producido sobre la base de un muestreo aleatorio de una población con varianzas iguales. Por lo tanto, la hipótesis nula de igualdad de varianzas se rechaza y se concluye que hay una diferencia entre las variaciones en la población.

Los resultados indican que **existen diferencias significativas** en las varianzas de los grupos creados por los valores de `uses_ad_boosts` y `badge_fast_shipping`. Para el resto de variables a estudiar, **no hay diferencias significativas** entre las varianzas de los grupos, es decir existe homogeneidad de varianza u homocedasticidad.

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.

```
print(corr_matrix)
```

##	estimate	p-value
## retail_price	0.54914958	6.556328e-84
## units_sold	0.07772473	1.167610e-02
## uses_ad_boosts	-0.10989721	3.555528e-04
## rating	0.06644395	3.116974e-02
## rating_count	0.19701440	1.150286e-10
## rating_five_count	0.20422930	2.280884e-11
## rating_four_count	0.20286444	3.112111e-11
## rating_three_count	0.18214944	2.668989e-09
## rating_two_count	0.16595843	6.164154e-08
## rating_one_count	0.17935725	4.684162e-09
## badges_count	0.02374221	4.417384e-01
## badge_local_product	0.06299565	4.106870e-02
## badge_product_quality	0.01130843	7.140967e-01
## badge_fast_shipping	0.01850128	5.488938e-01
## product_color	-0.02341265	4.481039e-01
## product_variation_inventory	0.38873242	2.789456e-39
## shipping_is_express	0.03751469	2.240789e-01
## countries_shipped_to	-0.02096194	4.970382e-01
## inventory_total	-0.05691424	6.499553e-02
## merchant_rating	0.06539757	3.393109e-02
## rating_interval	0.05781690	6.084722e-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 podemos concluir que, el coeficiente de correlación entre retail_price y price (0.5) 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 uno (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 μ_1 es la media de la población de la que se extrae la primera muestra y μ_2 es la media de la población de la que extrae la segunda. Así, tomaremos $\alpha = 0.05$.

```
t.test(SalesSummerObj.rating.interval.tres$price,SalesSummerObj.rating.interval.cinco$price,alternative = "less")

##
## Welch Two Sample t-test
##
## data: SalesSummerObj.rating.interval.tres$price and SalesSummerObj.rating.interval.cinco$price
## t = 0.32364, df = 229.22, p-value = 0.6267
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##      -Inf 0.8571741
## sample estimates:
## mean of x mean of y
##  7.837907  7.697455
```

Obtenemos un p-value mayor que el valor de significación, por lo que no podemos rechazar la hipótesis 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 `budget_product_quality` del producto es 1 y otra cuando dicha variable es 0***

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

```
t.test(SalesSummerObj.budget.productquality.cero$price,SalesSummerObj.budget.productquality.uno$price,alternative = "less")

##
## Welch Two Sample t-test
##
## data: SalesSummerObj.budget.productquality.cero$price and SalesSummerObj.budget.productquality.uno$price
## t = -0.20757, df = 65.383, p-value = 0.4181
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##      -Inf 0.8313407
## sample estimates:
## mean of x mean of y
##  8.244718  8.362833
```

Obtenemos un p-value mayor que el valor de significación, por lo que no podemos rechazar la hipótesis 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 0***

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

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

```
##
## Welch Two Sample t-test
##
## data: SalesSummerObj.uses.ad.boosts.cero$price and SalesSummerObj.uses.ad.boosts.uno$price
## t = 2.8237, df = 920.87, p-value = 0.9976
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##      -Inf 1.132499
## sample estimates:
## mean of x mean of y
##  8.570377  7.855011
```

Obtenemos un p-value mayor que el valor de significación, por lo que no podemos rechazar la hipótesis 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

Plantaremos 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
## -8.9427 -2.6216 -0.6216  2.2208 17.7059
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.581887   0.199335   28.00  <2e-16 ***
## retprice     0.042032   0.003723   11.29  <2e-16 ***
## pvinventory  0.054909   0.005037   10.90  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.575 on 1049 degrees of freedom
## Multiple R-squared:  0.2136, Adjusted R-squared:  0.2121
## F-statistic: 142.5 on 2 and 1049 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
## -8.6760 -2.6005 -0.5611  2.2167 18.0538
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.370255   0.204888  26.211  <2e-16 ***
## retprice     0.040723   0.003746  10.872  <2e-16 ***
## pvinventory  0.052578   0.005054  10.404  <2e-16 ***
## ratingfour   0.005207   0.005134   1.014   0.311
## ratingone    0.012510   0.007708   1.623   0.105
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.55 on 1047 degrees of freedom
## Multiple R-squared:  0.2264, Adjusted R-squared:  0.2234
## F-statistic: 76.59 on 4 and 1047 DF,  p-value: < 2.2e-16
```

Calidad del ajuste de los modelos

El Coeficiente de determinación R^2 mide el grado en el que el modelo de regresión lineal explica las 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 aparece al calcular ambos modelos con la función `lm()`, en nuestro caso es 0.2264 en el `modelo1`, pero conviene utilizar el ajustado que es: **0.2234**

Lo que implica que el `modelo1` es capaz de explicar alrededor del **22.34 % 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(0.2121)
r1
```

```
## [1] 0.4605432
```

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) 5.19074634 5.97302704
## retprice    0.03472716 0.04933729
## pvinventory 0.04502566 0.06479293
```

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 realmente mínima respecto a `retprice`.

En el segundo modelo comprobamos que el hecho de añadir los regresores `ratingfour` y `ratingone` mejora el coeficiente de determinación ajustado aunque mínimamente, que se queda en un 22.34%.

Realizaremos una predicción del precio de venta:

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

# Predecir el precio

predict(modelo1, newdata)
```

```
##          1
## 9.588318
```

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.

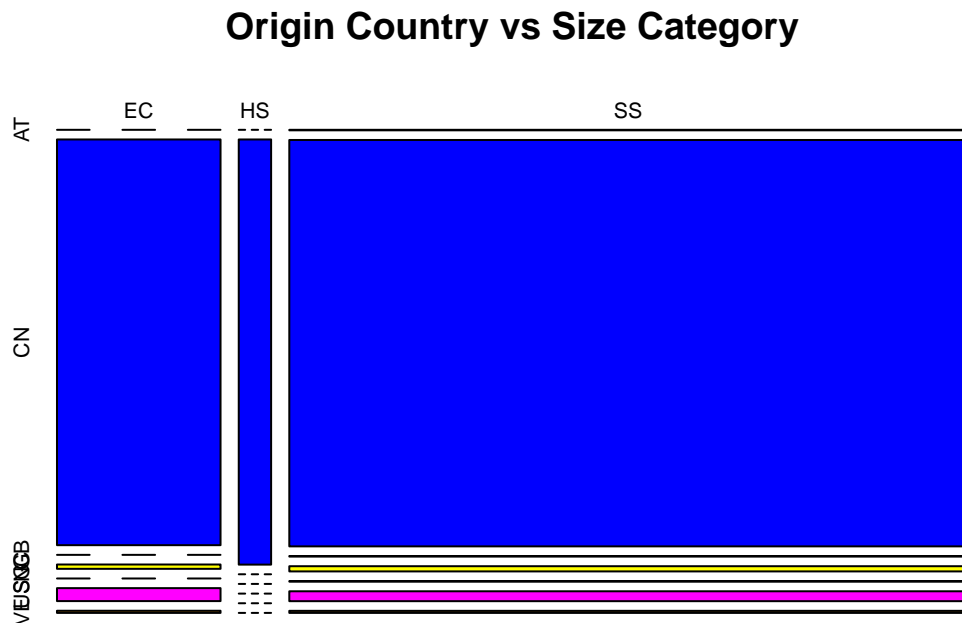
2.4.4.4.1 Tablas de Contingencia

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

```
Tabla.SC.OC
```

```
##
##      AT  CN  GB  NC  SG  US  VE
## EC   0 188   0   2   0   6   1
## HS   0  39   0   0   0   0   0
## SS   1 780   1  10   1  19   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.

```
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 = 2.9685, df = 12, p-value = 0.9958
```

```
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 = 0.9842  
## alternative hypothesis: two.sided
```

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

Cálculo del modelo

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

```
# 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 12.564006 16.728615 -5.904251  
## GB -6.920249 -2.284379 6.920151  
## NC 8.021308 -2.924827 -5.718242  
## SG -6.920249 -2.284379 6.920151  
## US 9.119283 -4.310894 -6.174407  
## VE 7.325471 -2.612316 -5.939105  
##  
## Std. Errors:  
## (Intercept) SalesSummerObj$size_categoryHS SalesSummerObj$size_categorySS  
## CN 38.9698400 3.384340e-06 38.9826651  
## GB 0.7252816 2.476857e-10 0.7225532  
## NC 38.9761825 4.656710e-08 38.9902715  
## SG 0.7252816 3.540281e-10 0.7225532  
## US 38.9719101 1.137314e-08 38.9853932
```

```
## VE 38.9826268 6.407023e-08 38.9986379
##
## Residual Deviance: 472.4827
## AIC: 508.4827
```

```
# Coeficientes Modelo
```

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

```
## (Intercept) SalesSummerObj$size_categoryHS SalesSummerObj$size_categorySS
## CN 12.564006 16.728615 -5.904251
## GB -6.920249 -2.284379 6.920151
## NC 8.021308 -2.924827 -5.718242
## SG -6.920249 -2.284379 6.920151
## US 9.119283 -4.310894 -6.174407
## VE 7.325471 -2.612316 -5.939105
```

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 cómo 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**:

$$\ln\left(\frac{P(Y = 1/X)}{1 - P(Y = 1/X)}\right) = 12.564 + 16.729 * HS - 5.904 * SS$$

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:

$$\ln\left(\frac{P(Y=1/X)}{1-P(Y=1/X)}\right) = 9.112 - 4.31 * HS - 6.174 * SS$$

Odds Ratios Modelo

```
exp(coefmodel.sizecategory.origincountry)
```

```
##      (Intercept) SalesSummerObj$size_categoryHS SalesSummerObj$size_categorySS
## CN  2.860740e+05      1.841388e+07      2.727823e-03
## GB  9.875841e-04      1.018372e-01      1.012472e+03
## NC  3.045158e+03      5.367396e-02      3.285483e-03
## SG  9.875841e-04      1.018372e-01      1.012472e+03
## US  9.129652e+03      1.342154e-02      2.082040e-03
## VE  1.518489e+03      7.336442e-02      2.634386e-03
```

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) -63.81548 88.94349
## SalesSummerObj$size_categoryHS 16.72861 16.72862
## SalesSummerObj$size_categorySS -82.30887 70.50037
##
## , , GB
##
##              2.5 %    97.5 %
## (Intercept) -8.341775 -5.498723
## SalesSummerObj$size_categoryHS -2.284379 -2.284379
## SalesSummerObj$size_categorySS 5.503972 8.336329
##
## , , NC
##
##              2.5 %    97.5 %
```



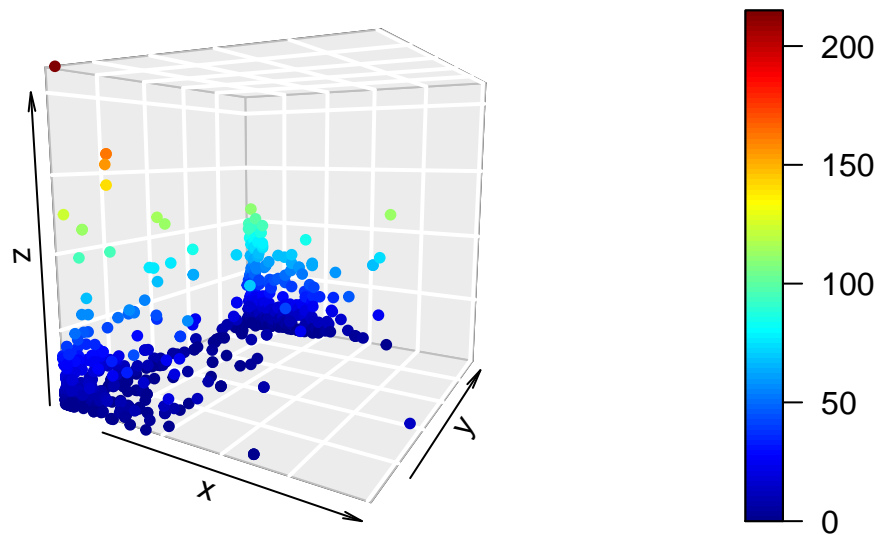
```
## (Intercept) -68.370606 84.413222
## SalesSummerObj$size_categoryHS -2.924827 -2.924827
## SalesSummerObj$size_categorySS -82.137770 70.701286
##
## , , SG
##
## 2.5 % 97.5 %
## (Intercept) -8.341775 -5.498723
## SalesSummerObj$size_categoryHS -2.284379 -2.284379
## SalesSummerObj$size_categorySS 5.503972 8.336329
##
## , , US
##
## 2.5 % 97.5 %
## (Intercept) -67.264257 85.502823
## SalesSummerObj$size_categoryHS -4.310894 -4.310894
## SalesSummerObj$size_categorySS -82.584374 70.235559
##
## , , VE
##
## 2.5 % 97.5 %
## (Intercept) -69.079074 83.730015
## SalesSummerObj$size_categoryHS -2.612316 -2.612316
## SalesSummerObj$size_categorySS -82.375031 70.496821
```

2.5 - REPRESENTACIÓN DE RESULTADOS

Interpretación de Modelos

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

```
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 bajos 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, McNemar y Q-Cochran: [https://www.cienciadedatos.net/documentos/22.2_test_exacto_de_fisher_chi-cuadrado_de_pearson_mcnemar_qcochran#\(chi%5E2\)_de_Pearson_\(test_de_independencia\)](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 goodness 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/modelosderespuestamultinomialconraplicacionparaelestudiodeladepr>

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