# Webscraping en amazon

Anthony J. Servita R.

12/12/2020

1. **- conocimiento del problema.** Se realizara una extraccion de datos (webscraping) sobre la pagina web de amazon, se hara una revision de los porductos computacionales que posee amazon en venta. se extraeran los modelos, las marcas asi como su precio y fecha de publicacion.

importación de paquetes para la extraccion de datos.

```
library(rvest)
```

```
## Loading required package: xml2
```

```
library(robotstxt)
library(selectr)
library(xml2)
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.0 ──
```

```
## ✓ ggplot2 3.3.2     ✓ purrr   0.3.4
## ✓ tibble  3.0.4     ✓ dplyr   1.0.2
## ✓ tidyr   1.1.2     ✓ stringr 1.4.0
## ✓ readr   1.4.0     ✓ forcats 0.5.0
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## x dplyr::filter()         masks stats::filter()
## x readr::guess_encoding() masks rvest::guess_encoding()
## x dplyr::lag()            masks stats::lag()
## x purrr::pluck()          masks rvest::pluck()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

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

```
library(dplyr)
library(skimr)
library(tidyr)
library(readr)
```

**2.- Extraccion de datos.** El webscraping se realizara por medio del paquete rvest.

```
# vecto de caracter que contiene el template de amazon a la cual estoy accediendo
url <-paste("https://www.amazon.com/-/es/s?i=computers-intl-ship&bbn=16225007011&rh=n%3A1
6225007011%2Cn%3A13896617011%2Cn%3A565098&dc&language=es&qid=1607804959&rnid=2057412011&r
ef=sr_nr_p_n_feature_fourteen_browse-bin_2")

#asistencia del robots.txt para ver si esta disponible la extraccion de datos legalmente
paths_allowed(paths = url)

##  www.amazon.com

## [1] TRUE
```

Dado que el valor del robot.txt nos arrojo un TRUE, extrearemos los datos que queremos de la pagina objetivo.

```
#instancia que responde a la extraccion del code HTML de la pagina objetivo.
pagina_web <- read_html(url)

# vector de la clase css que situa el nombre del producto.
css_producto <- c(".s-line-clamp-2")

# se extrae en CODE HTML que contiene el nombre del producto
producto_html <- html_nodes(pagina_web, css_producto)

# Extreccion de nombre del producto
producto_texto <- html_text(producto_html)

# limpieza de cadenas de texto sobre los productos
producto_texto <- gsub("\n", "", producto_texto)
producto_texto <- gsub(" ", "", producto_texto)







nombre_col <- c("PC_producto",
                "procesador",
                "discoduro",
                "RAM",
                "artefactos",
                "tarjetaswifi")
producto_texto <- read_delim(producto_texto,
                             ",", col_names = F)

## Warning: 12 parsing failures.
## row col  expected    actual         file
##   2  -- 6 columns 1 columns literal data
##   4  -- 6 columns 8 columns literal data
##   5  -- 6 columns 7 columns literal data
##   6  -- 6 columns 7 columns literal data
##   7  -- 6 columns 9 columns literal data
## ... ... ......... ......... ............
## See problems(...) for more details.
```

```r
colnames(producto_texto) <- nombre_col
head(producto_texto)
```

```
## # A tibble: 6 x 6
##   PC_producto         procesador     discoduro  RAM    artefactos tarjetaswifi
##   <chr>               <chr>          <chr>      <chr>  <chr>      <chr>
## 1 CyberpowerPCGamerX… Inteli5-10400… GeForceGT… 8GBDD… 500GBNVMe… WiFiReady&Win…
## 2 iBUYPOWERTrace-Com… <NA>           <NA>       <NA>   <NA>       <NA>
## 3 CyberpowerPCGamerS… AMDRyzen73800… RadeonRX5… 16GBD… 1TBNVMeSSD WiFi&Win10Hom…
## 4 AcerAspireTC-895-U… procesadorInt… 12GB2666M… 512GB… 8XDVD      802.11axWi-Fi6
## 5 SkytechArchangelGa… GTX1660Super6G 500GBSSD   16GBD… ventilado… Windows10Home…
## 6 SkyTechBlazeII-Com… NVIDIAGeForce… 500GSSD    8GBDD… RGB        ACWiFi
```

```r
#obtenemos los precios
css_precio <- c(".a-price-whole")
precio_html <- html_nodes(pagina_web, css_precio)
precio_texto <- html_text(precio_html)

#limpieza de las cadenas de precio
precio_texto <- gsub(",", "", precio_texto)

#visualizacion de los precios
head(precio_texto)
```

```
## [1] "859."  "699."  "1229." "529."  "999."  "829."
```

### 3.- Transformacion de variables

```r
# Dataframe con las variables extraidas
df <- data.frame(producto_texto,
                 "precio" = as.numeric(precio_texto))



write.csv(df, file = "tabla de datos-WS-PC.csv")
```

luego de realizar dla extraccion de datos y arreglarlos, se importaran estos datos para realizar un analisis descriptivo.

### 4.- importacion de newdata.

```r
tabla_de_datos_WS_PC <- read_delim("tabla de datos-WS-PC1.csv",
    ";", escape_double = FALSE, na = "0",
    trim_ws = TRUE)
```

```
##
## ── Column specification ────────────────────────────────────────────────────────
## cols(
##   PC_producto = col_character(),
##   procesador = col_character(),
##   discoduro = col_character(),
##   RAM = col_character(),
##   caracteristicas = col_character(),
##   tarjetaswifi_OS = col_character(),
##   precio = col_double()
## )
```

```r
head(tabla_de_datos_WS_PC)
```

```
## # A tibble: 6 x 7
##   PC_producto  procesador discoduro RAM   caracteristicas tarjetaswifi_OS precio
##   <chr>        <chr>      <chr>     <chr> <chr>           <chr>             <dbl>
## 1 CyberpowerP… "Inteli5-… "GeForce… "8GB… "500GBNVMeSSD"  "WiFiReady&Win…    791
## 2 iBUYPOWERGa… "Inteli7-… "1TBHDD … "16G… "NVIDIAGTX1660… "Wi-Filisto"      1275
## 3 iBUYPOWERTr… ""         ""        ""    ""              ""                 699
## 4 AcerAspireT… "procesad… "512GBNV… "12G… "8XDVD"         "802.11axWi-Fi…    529
## 5 CyberpowerP… "AMDRyzen… "1TBNVMe… "16G… "RadeonRX5700X… "WiFi&Win10Hom…   1229
## 6 SkytechArch… "Ryzen536… "500GBSS… "16G… "ventiladoresR… "Windows10Home…    999
```

**5.- conocimiento de los datos.** Los datos que se han importado, estan contenidos en una tabal de 16 x 7 los cuales contienen valores para las 5 variables cualitativas y 1 variable cuantitativa.

```
df <- tabla_de_datos_WS_PC
glimpse(df)
```

```
## Rows: 16
## Columns: 7
## $ PC_producto     <chr> "CyberpowerPCGamerXtremeVRGamingPC", "iBUYPOWERGaming…
## $ procesador      <chr> "Inteli5-10400F2.9GHz", "Inteli7-10700F2.9GHz", "", "…
## $ discoduro       <chr> "GeForceGTX1660Super6GB", "1TBHDD y 240GBSSD", "", "5…
## $ RAM             <chr> "8GBDDR4", "16GBDDR4RAM", "", "12GB2666MHzDDR4", "16G…
## $ caracteristicas <chr> "500GBNVMeSSD", "NVIDIAGTX1660Ti6GB", "", "8XDVD", "R…
## $ tarjetaswifi_OS <chr> "WiFiReady&Win10Home(GXiVR8060A10)", "Wi-Filisto", ""…
## $ precio          <dbl> 791, 1275, 699, 529, 1229, 999, 829, 449, 579, 684, 9…
```

Al parecer el tipo de datos precio se guardo por valores double, se realizar un cambio del tipo de variable, corrigiendo este valor para evitar problema a futuro.

```
df <- df %>%
  mutate(precio = as.integer(precio))
glimpse(df)
```

```
## Rows: 16
## Columns: 7
## $ PC_producto     <chr> "CyberpowerPCGamerXtremeVRGamingPC", "iBUYPOWERGaming…
## $ procesador      <chr> "Inteli5-10400F2.9GHz", "Inteli7-10700F2.9GHz", "", "…
## $ discoduro       <chr> "GeForceGTX1660Super6GB", "1TBHDD y 240GBSSD", "", "5…
## $ RAM             <chr> "8GBDDR4", "16GBDDR4RAM", "", "12GB2666MHzDDR4", "16G…
## $ caracteristicas <chr> "500GBNVMeSSD", "NVIDIAGTX1660Ti6GB", "", "8XDVD", "R…
## $ tarjetaswifi_OS <chr> "WiFiReady&Win10Home(GXiVR8060A10)", "Wi-Filisto", ""…
## $ precio          <int> 791, 1275, 699, 529, 1229, 999, 829, 449, 579, 684, 9…
```

```
skim(df)
```

*Data summary*

| | |
|---|---|
| Name | df |
| Number of rows | 16 |
| Number of columns | 7 |

_____

Column type frequency:

| | |
|---|---|
| Character | 6 |
| Numeric | 1 |

| | |
|---|---|
| Group variables | None |

**Variable type: character**

| skim_variable | n_missing | complete_rate | min | max | empty | n_unique | whitespace |
|---|---|---|---|---|---|---|---|
| PC_producto | 0 | 1 | 15 | 53 | 0 | 15 | 0 |
| procesador | 0 | 1 | 0 | 65 | 1 | 16 | 0 |
| Discoduro | 0 | 1 | 0 | 23 | 1 | 14 | 0 |
| RAM | 0 | 1 | 0 | 15 | 1 | 11 | 0 |
| caracteristicas | 0 | 1 | 0 | 34 | 2 | 14 | 0 |
| tarjetaswifi_OS | 0 | 1 | 0 | 33 | 4 | 12 | 0 |

**Variable type: numeric**

| skim_variable | n_missing | complete_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
|---|---|---|---|---|---|---|---|---|---|---|
| Precio | 0 | 1 | 937.88 | 628.23 | 236 | 579.75 | 745 | 1056.5 | 2922 | ▇▆▁▁▁ |

*Conclusiones.* Los precios que maneja amazon sobre la muestra extraida es de 938 USD por computador, con una desviacion de 628 USD. Asi mismo, el precio mas bajo por computador equivale a los 236 USD y el mas alto contiene un valor de 2922 USD. hasta el 50% de los datos, los precios llegan hasta 745 USD. Sobre el histograma podemos visualizar que coeficiente de asimetria es alto dado que su asimetria es positiva. Ergo, el valor mas frecuente es menor a la media y a la mediana.

**Agruparemos los precio por computador.**

```
df %>%
  select(PC_producto, precio) %>%
  group_by(PC_producto)
```

```
## # A tibble: 16 x 2
## # Groups:   PC_producto [15]
##    PC_producto                                          precio
##    <chr>                                                 <int>
##  1 CyberpowerPCGamerXtremeVRGamingPC                       791
##  2 iBUYPOWERGamingPCDesktopElementMR9320                  1275
##  3 iBUYPOWERTrace-Computadoragamerdeescritorio             699
##  4 AcerAspireTC-895-UA92Desktop                            529
##  5 CyberpowerPCGamerSupremeLiquidCoolGamingPC             1229
##  6 SkytechArchangelGamingComputerPCDesktop                 999
##  7 SkyTechBlazeII-Computadoradeescritorioparavideojuegos   829
##  8 HP22PCtodoenuno                                         449
##  9 AcerAspireC24-963-UA91AIODesktop                        579
```

```
## 10 LenovoIdeaCentreAIO3                                  684
## 11 SkytechShivaGamingPCDesktop                           999
## 12 OMEN30LGamingDesktopPC                               2922
## 13 DellInspironDesktop3880                               649
## 14 AcerChromeboxCXI3-UA91MiniPC                          236
## 15 NuevosobremesaparajuegosAlienwareAuroraR10           1557
## 16 LenovoIdeaCentreAIO3                                  580
```

*interpretacion.* Nos damos cuenta que el computador mas barato es una miniPC marca Acer chromeboxCXI3 con un valor en USD de 236. el mas costoso una PC de escritorio para gaming (juegos) valorada en 2922 USD.

agrupemos los precios por procesador de computadora.

```
df %>%
  select(procesador, precio) %>%
  group_by(procesador)
```

```
## # A tibble: 16 x 2
## # Groups:   procesador [16]
##    procesador                                                    precio
##    <chr>                                                          <int>
##  1 "Inteli5-10400F2.9GHz"                                           791
##  2 "Inteli7-10700F2.9GHz"                                          1275
##  3 ""                                                               699
##  4 "procesadorIntelCorei5-10400de6nucleos"                          529
##  5 "AMDRyzen73800X3.9GHz"                                          1229
##  6 "Ryzen536003.6GHz"                                               999
##  7 "Ryzen526006nucleos3.4GHz"                                       829
##  8 "procesadorAMDAthlonGold3150U"                                   449
##  9 "IntelCorei3-1005G1"                                             579
## 10 "\"24\"\"All-in-OneOrdenadorTodo-en-UnoAMDRyzen54500UProcesadorMóvil\…    684
## 11 "AMDRyzen52600"                                                 999
## 12 "procesadorIntelCorei9-10850K"                                 2922
## 13 "IntelCorei510thGen"                                            649
## 14 "IntelCeleron3867UProcessor1.8GHz"                              236
## 15 "AMDRyzen73700X y AMDRadeonRX5700XT"                           1557
## 16 "AMDRyzen34300UMobileProcessor"                                 580
```

*interpretacion.* Los computadores mas caros contienen un procesador intel core i9; en contraste, el computador mas barato posee un procesador intel celeron de 1.8GHZ de frecuencia.

Agrupemos por memoria RAM.

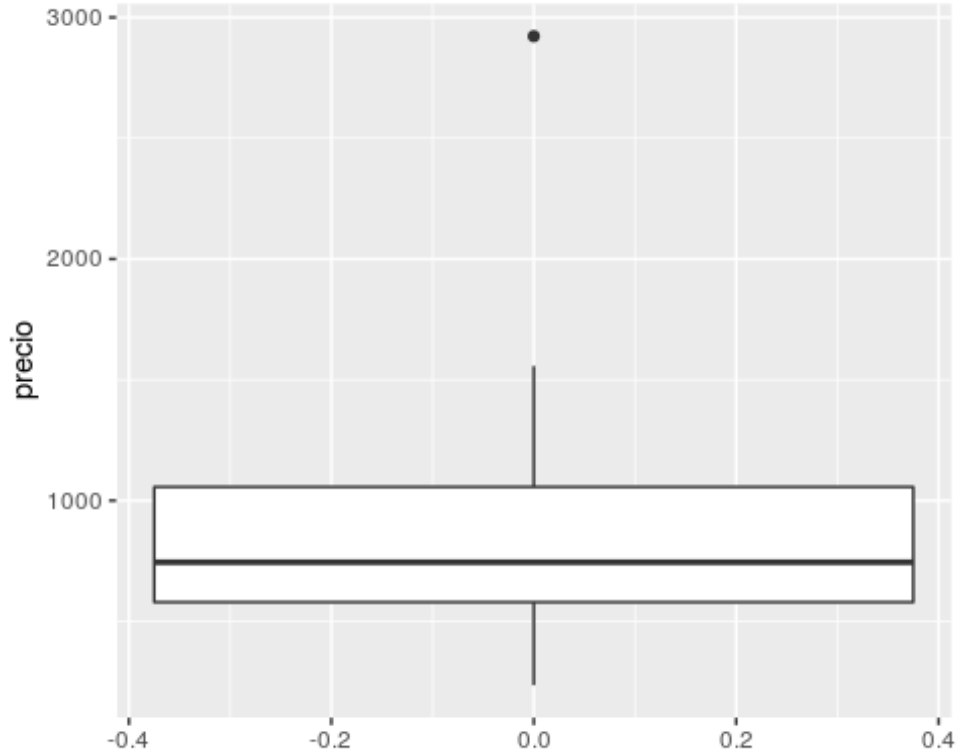
```
df %>%
  select(RAM, precio) %>%
  group_by(RAM)
```

```
## # A tibble: 16 x 2
## # Groups:   RAM [11]
##    RAM               precio
##    <chr>              <int>
##  1 "8GBDDR4"            791
##  2 "16GBDDR4RAM"       1275
##  3 ""                   699
##  4 "12GB2666MHzDDR4"    529
##  5 "16GBDDR4"          1229
```

```
##  6 "16GBDDR43000MHz"    999
##  7 "8GBDDR4"            829
##  8 "4GBdeRAM"           449
##  9 "8GBDDR4"            579
## 10 "16GBDDR4"           684
## 11 "16GBDDR4"           999
## 12 "32GBRAM"           2922
## 13 "12GBMemory"         649
## 14 "4GBDDR4-Memory"     236
## 15 "8GBGDDR6 y 16GB"   1557
## 16 "8GBDDR4"            580
```

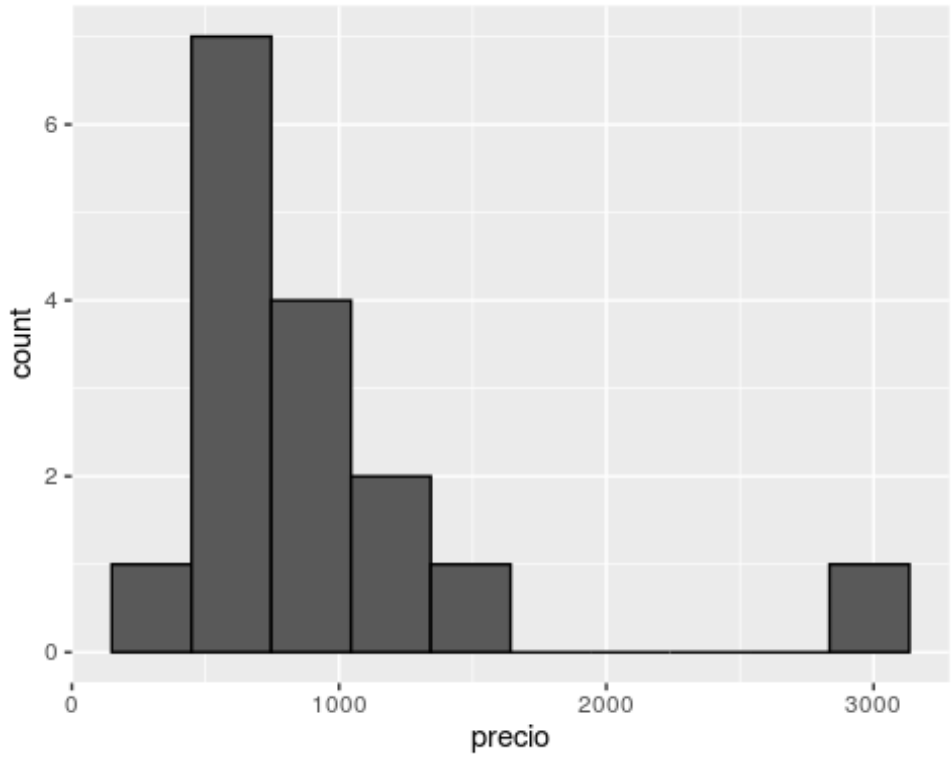
interpretacion. el computador mas caro, viene con una memoria RAM de 32GB. el mas barato contiene una RAM de 4GBDDR4.

6. **Analisis de datos pro graficos.**




*interpretacion grafica.* solo encontamos un valor perdido sobre el grafico de boxplot. el rango intercuartilico se encuentra por debajo de los 2000 USD por lo que los precio de los computadores no exceden los 2000 USD.

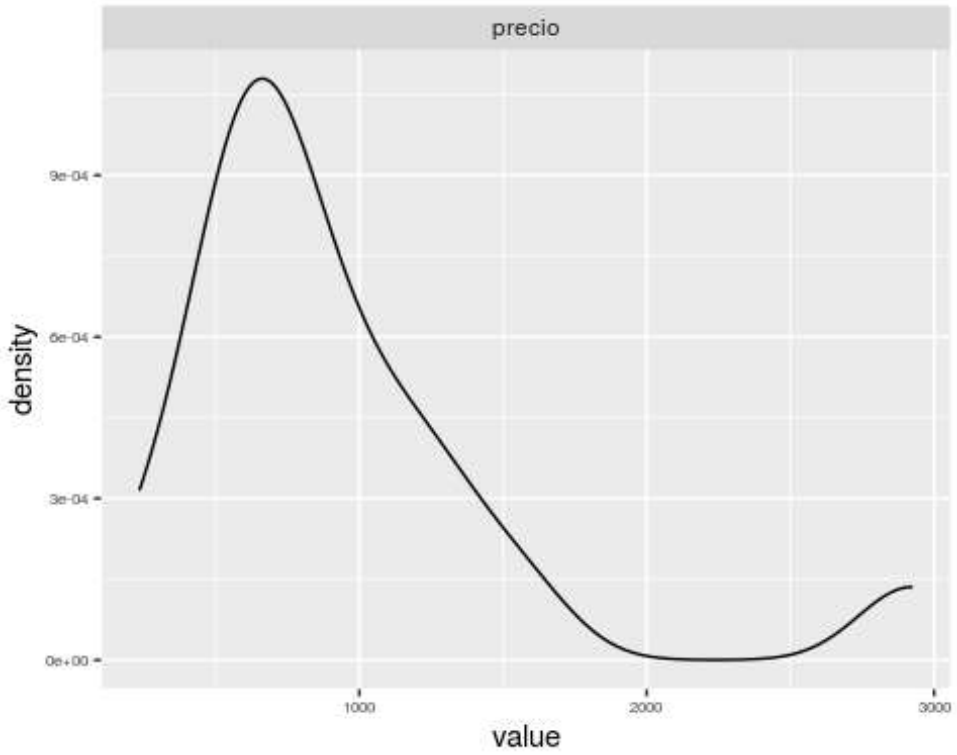
```
ggplot(df) + geom_histogram(aes(precio), bins = 10, colour = 'black')
```




Este acercamiento al histograma nos muestra que el valor promedio es menor al valor de la moda, aun asi el la asimetria es positiva.

## 7. Analisis exploratorio de varibles (EDA).

```
df %>%
  select_if(is.integer) %>%
  gather() %>%
  ggplot(aes(value)) + geom_density() +
  facet_wrap(~key, scales = 'free') +
  theme(axis.text = element_text(size = 6))
```




Con este Grafico visualizamos mejor la densidad de los datos, y su distribucion. podemos ver que no existe una distribucion normal.

Aplicaremos el test de normalidad de shapiro wilk para corroborar la normalidad de los datos. para esto, se contrastaran la hipotesis nula de ditribucion normal sobre los datos con un nivel de significancia del 5%.

```
shapiro.test(df$precio)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$precio
## W = 0.77574, p-value = 0.00132
```

Conclusiones. Dado que el valor de probabilidad es inferior al valor de significancia, de 0,05. entonces, no existe evidencia suficiente para no rechazar la hipotesis nula, por lo que se tiene un 95% de confianza que para esta muestra los datos no se encuentran distribuidos normalmente.

**8.- Analisis inferencial**. En el siguiente analisis se realizara una inferencia sobre el promedio de la poblacion de los precio que se encuentran reunidos en el pagina de amazon sobre la ventas de computadores.

para un nivel de significancia de 5%, y una desviacion estandar de 628USD.

```
length(df$precio)
```

```
## [1] 16
```

```
#ratio de la desviacion y la raiz de n
ratio <- sd(df$precio)/sqrt(length(df$precio))
#calcular t-student dado qeu los datos no poseen distribucion normal.
valor_t <- qt(c(1-(0.05/2)), c(16-1))
```

```r
Rempirica_95 <- c(ratio * valor_t)
lim_inf <- c(mean(df$precio) - Rempirica_95)
lim_sup <- c(mean(df$precio) + Rempirica_95)
inter_construction <- c(lim_inf, lim_sup)
print(inter_construction)
```

```
## [1]  603.1173 1272.6327
```

*Conclusiones.* Con un 95% de confianza inferimos que el valor del promedio para el precio de los computadores se encuentra entre los 603.1173 y 1272.6327 USD por computadora, dado los componentes que la misma proporciona.