Tipología y ciclo de vida de los datos

PRA 1: Web scraping

David Poggio Moro

Contenido

1. Contexto

1.1. Descripción

Para esta práctica nos planteamos el proyecto para investigar sobre la web de venta online de tecnología PcComponentes.com ya que, por un lado, me considero aficionado personal de tecnología que asiduamente adquiere productos de este eCommerce y, por otro, que al igual que yo hay miles de personas que toman esta web como referencia a la hora de precios y estado del mercado respecto a tecnología ya que es uno de los principales referentes en España en este ámbito.

Dentro de esta web podemos encontrar una enorme variedad de categorías de productos que no sólo se venden tal cual vienen del proveedor, sino que también tienen un apartado de outlet en el que se venden productos reacondicionados a cambio de una reducción de precio. Nos centraremos en este último ya que se podrían sacar conclusiones interesantes como qué tipo de producto tiende a aparecer más en esta sección o cuáles son los productos que obtienen un mayor descuento.

1.2. Análisis previo

Realizaremos un pequeño análisis previo en el que repasaremos las características principales del sitio web. Continuaremos con nuestro análisis puesto que se ha leído previamente el aviso legal que se encuentra enlazado en el pie de página de la web y en el que vemos que mientras evitemos daños a los sistemas de la web no tendremos ningún problema.

1.2.1. Robots.txt

Si accedemos al archivo de robots del sitio (que hemos subido al repositorio para tener una versión estática del momento en el que hacemos la práctica por si cambiara en el futuro).

Observando este fichero vemos que la url específica que queremos utilizar no está bloqueada por lo que no tendremos problemas a priori por este aspecto.

Si notamos que no se bloquean todos los bots que puedan ir contra la web sino que se hace en función de user agent como los de Baidu o Yandex ya que estos spiders al estar basados en China y Rusia respectivamente no están dentro de los países a los que pccomponentes vende y por tanto sería un desperdicio de recursos permitirles que indexen su web.

También podemos ver que a ciertos user agents no se les bloquea como tal sino que se les da un tiempo mínimo entre llamadas.

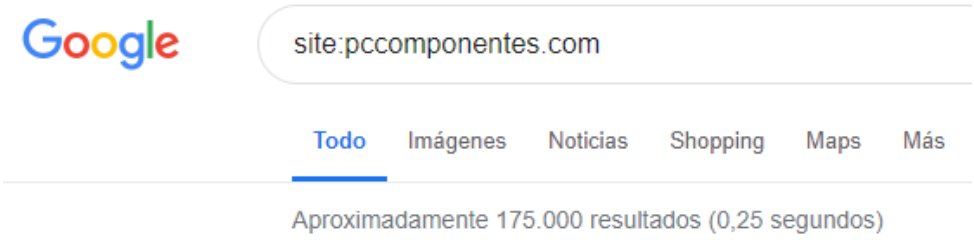
1.2.2. Sitemap.xml

Desde su web también podemos acceder al sitemap (que también hemos subido al repositorio por las mismas razones que antes) podemos ver que este, a su vez, apunta a otros sub mapas dependiendo del apartado como las categorías, los artículos o las marcas.

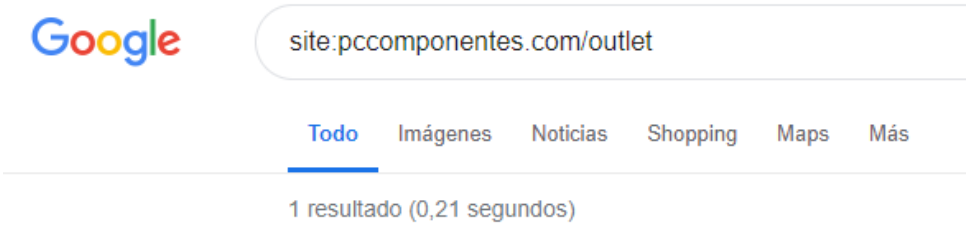
No nos interesa profundizar en estos submapas puesto que vamos a analizar una única url donde tendremos toda la información que nos interesa.

1.2.3. Tamaño

Una forma para analizar el tamaño del sitio es hacer una búsqueda en Google en el que utilicemos el parámetro "site:". En este caso utilizaríamos "site:pccomponentes.com" y Google nos ofrecería el siguiente resultado:



Donde podemos ver que tienen aproximadamente 175.000 sitios dentro de la web. De todos ellos, como hemos dicho antes, nosotros centraremos nuestro análisis en solo uno de ellos ya que el contenido de la página se puede ver todo en la misma url sin necesidad de paginación para mostrar el resto de artículos. Prueba de ello es que, si cambiamos la búsqueda en Google a nuestra url, obtenemos un solo resultado:



1.2.4. Tecnologías

Para realizar el análisis de tecnologías nos vamos a basar en la herramienta Builtwith de Python. En el repositorio se puede consultar el pequeño script tecnologias.py que hemos creado para obtener el json resultante:

```
{
    "cdn":[
        "CloudFlare"
    ],
    "font-scripts":[
        "Google Font API"
    ],
    "tag-managers":[
        "Google Tag Manager"
    ],
    "javascript-
frameworks":[
        "Prototype",
        "RequireJS",
        "jQuery"
    ],
    "web-frameworks":[
        "Twitter Bootstrap"
    ]
}
```

Lamentablemente, con nuestros conocimientos actuales no identificamos nada de este stack de tecnologías que afecte al ejercicio que vamos a realizar.

1.2.5. Whois

Mediante el análisis del archivo whois podemos identificar información sobre el propietario del sitio web. Mediante el script propietario.py que hemos creado en el repositorio podemos obtener esta información:


```
{
      "domain_name": [
            "PCCOMPONENTES.COM",
            "pccomponentes.com"
      ],
      "registrar": "Hosting Concepts B.V. d/b/a
Openprovider",
      "whois_server": "whois.registrar.eu",
      "referral_url": null,
      "updated_date": [
            "2015-12-09 10:42:36",
            "2015-10-23 06:05:32"
      ],
      "creation_date": "2004-11-20 10:12:21",
      "expiration_date": "2025-11-20 10:12:21",
      "name_servers": [
            "PETE.NS.CLOUDFLARE.COM",
            "VERA.NS.CLOUDFLARE.COM"
      ],
      "status": "clientTransferProhibited
https://icann.org/epp#clientTransferProhibited",
      "emails": "abuse@registrar.eu",
      "dnssec": "unsigned",
      "name": "REDACTED FOR PRIVACY",
      "org": "PC Componentes y Multimedia SL",
      "address": "REDACTED FOR PRIVACY",
      "city": "REDACTED FOR PRIVACY",
      "state": "Murcia",
      "zipcode": "REDACTED FOR PRIVACY",
      "country": "ES"
}
```

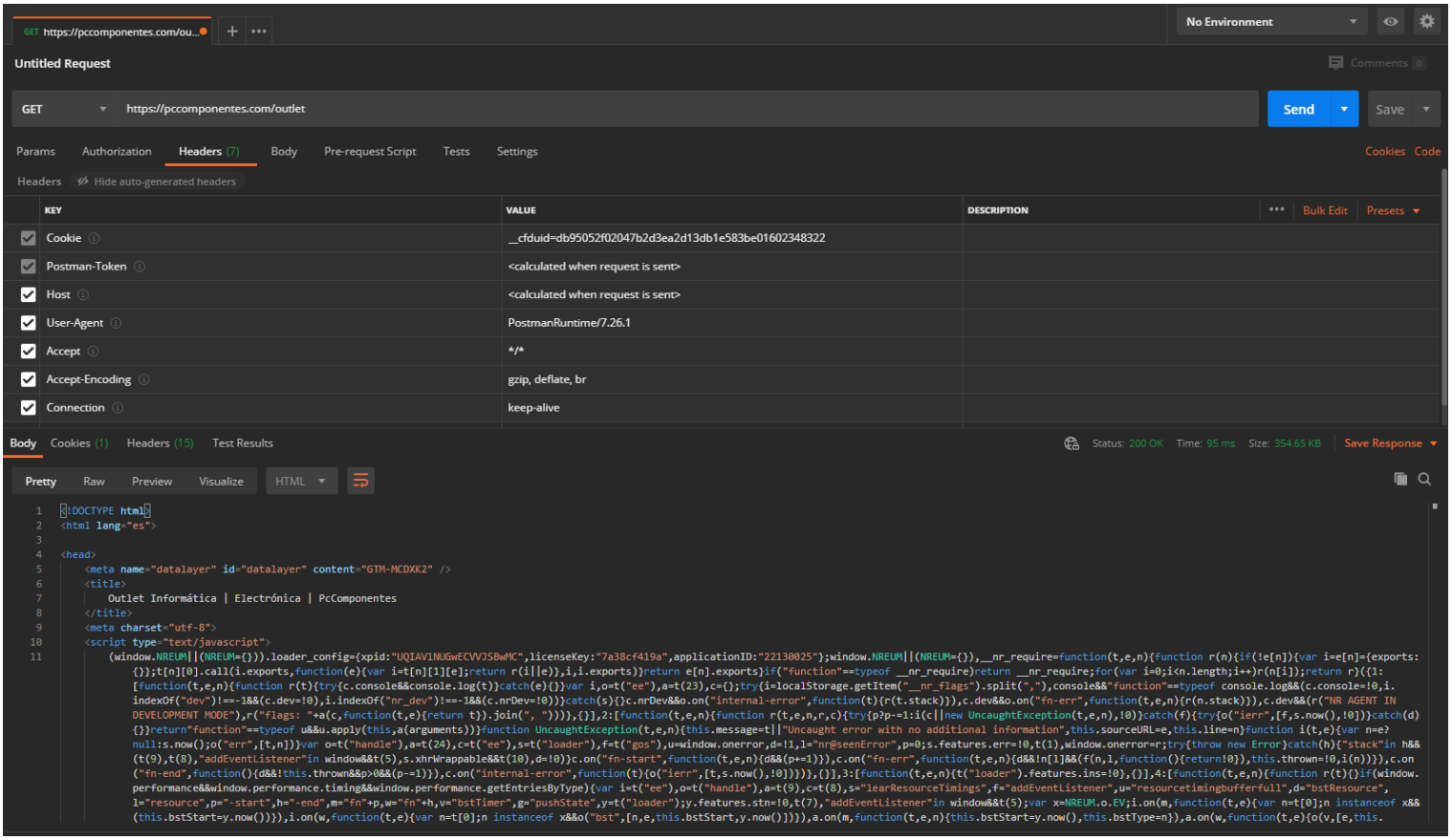

Los únicos datos que conseguimos sobre el propietario son la Sociedad Limitada a la que está registrada y que está en Murcia (España). El resto de información parece estar censurada por privacidad. Si quisiéramos el resto de información posiblemente tendríamos que ponernos en contacto con CloudFare puesto que ellos están obligados a tener la información según el ICANN.

1.3. Creación del dataset

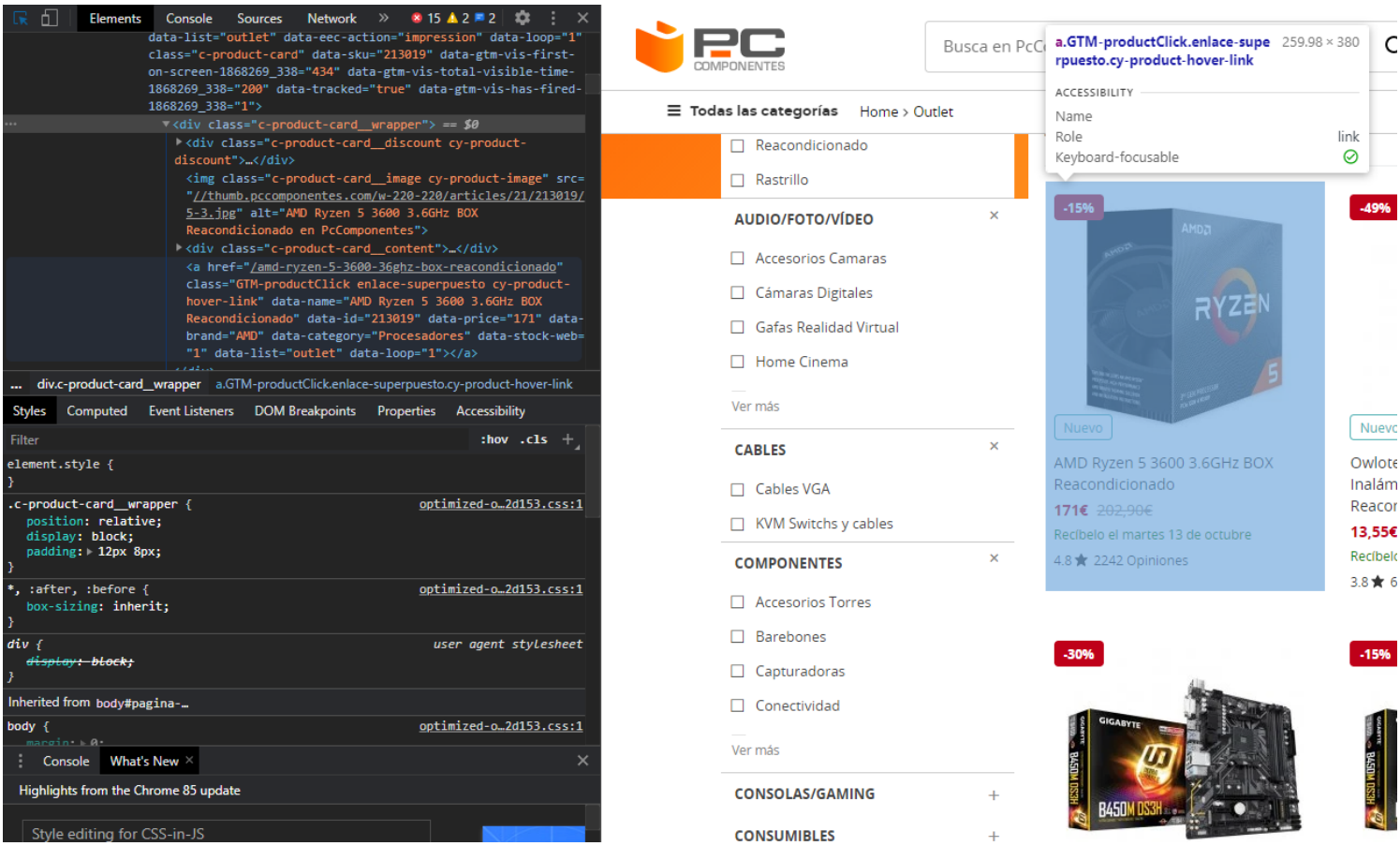
Una vez realizado el análisis previo ya estamos capacitados para realizar nuestro scaping. Para esto tendremos primero que identificar los parámetros con los que realizaremos nuestra llamada a la web y los tag que querremos capturar durante el proceso para volcarlos en el dataframe.

1.3.1. Identificación de parámetros

Mediante el programa Postman realizamos una petición GET a la URL que tenemos como objetivo para ver las cabeceras que necesitamos. De primeras dejamos todas las cabeceras por defecto y vemos que obtenemos la respuesta con el HTML de la web:



Ahora que hemos conseguido una respuesta, nos será más sencillo analizar este HTML directamente desde el navegador gracias a las DevTools que integra Google Chrome en este caso. Con ellas podemos seleccionar con el ratón una parte de la web y ver inmediatamente en qué parte del código se encuentra:



Gracias a esto podemos ver que todos los artículos están dentro de un tag article con la clase "c-product-card"

De aquí podemos obtener la siguiente información en los siguientes atributos:

- Nombre del producto: data-name
- Precio rebajado: data-price
- Marca: data-brand
- Categoría: data-category
- Stock: data-stock-web
- Identificador del producto: data-sku

Por otro lado, para obtener el precio original antes del descuento por reacondicionamiento tenemos que ir hasta un tag span dentro de un div con la clase "c-product-card__prices-actual c-product-card__prices-actual--discount cy-product-price-discount" que a su vez está dentro del div con clase "c-product-card__prices cy-product-price", dentro del div con clase "c-product-card__content" que, por último, está dentro del div con clase "c-product-card__wrapper".

Con estos datos, además calcularemos también el tanto por ciento de descuento que se está aplicando a cada producto.

1.3.2. Ejecución

Tras toda la información analizada nos encontramos preparados para hacer un script en Python que ejecute las llamadas necesarias y procese la información para poder exportarla cómodamente en un csv tal como nos pide el enunciado.

Este código lo hemos subido al repositorio como scrap.py. El código ha sido comentado para explicar lo que se pretende con cada bloque.

Como resultado de esta ejecución hemos obtenido el DOI y lo hemos extraído en el fichero Dataset.csv que hemos subido también al repositorio.

2. Título

El título que elegimos para este dataset es “Descuentos del outlet de PCComponentes”.

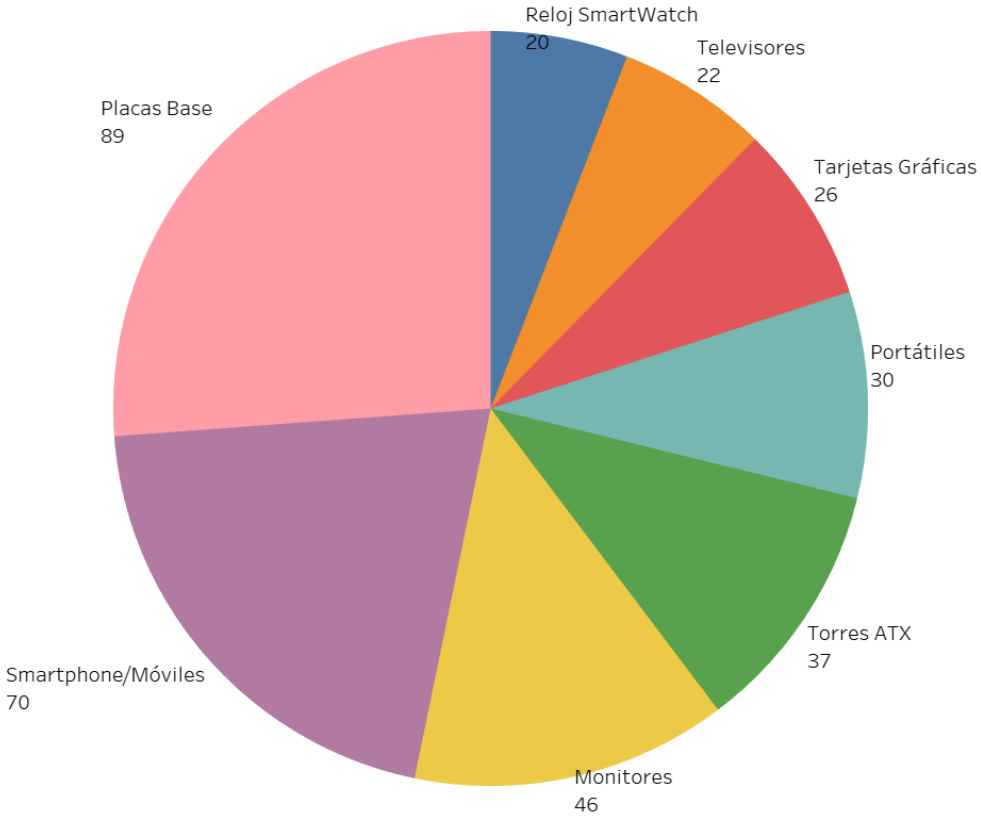
3. Descripción del dataset

En este dataset pretendemos obtener los datos de los descuentos que hay en la sección de outlet de PCComponentes.

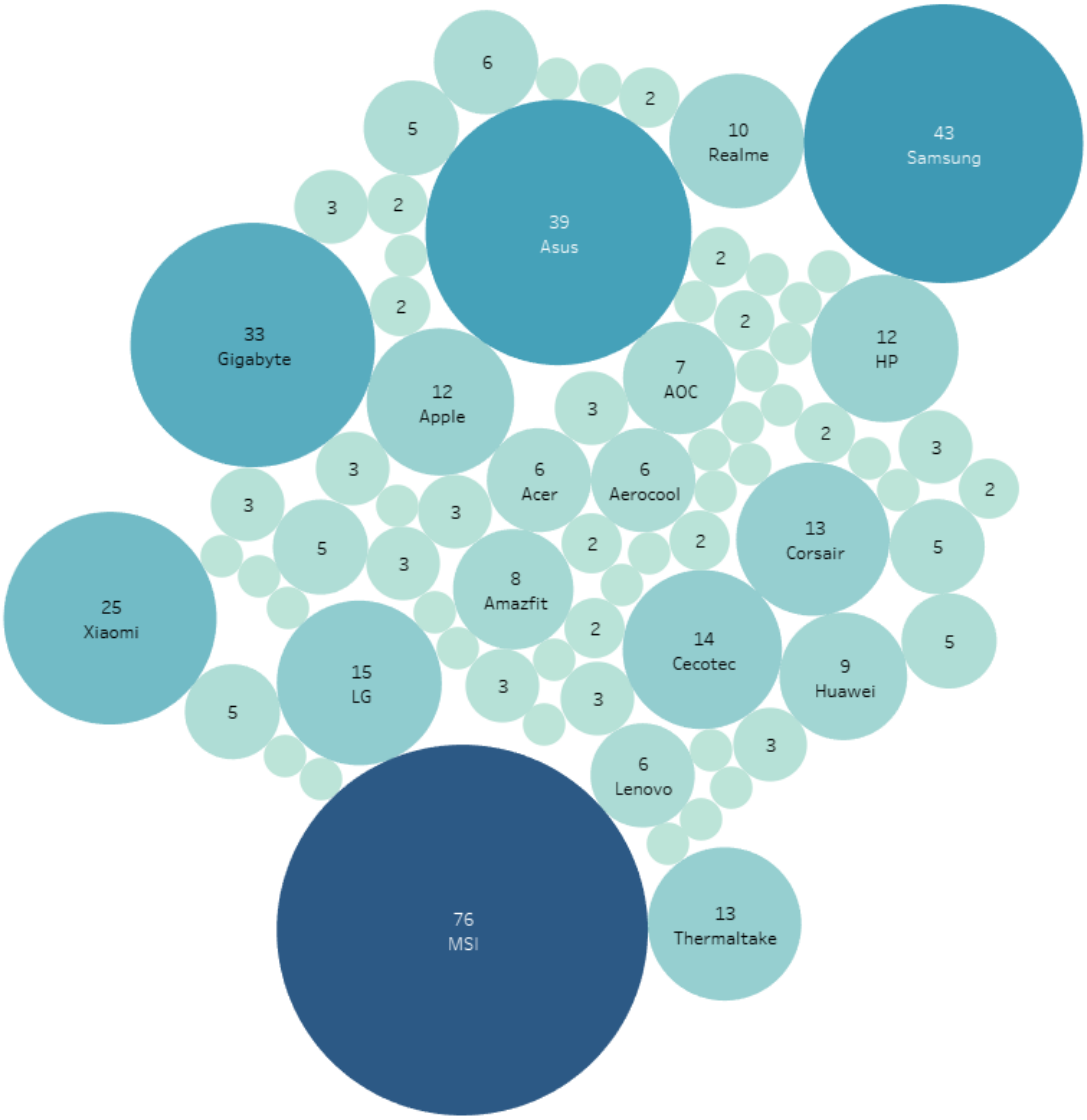
4. Representación gráfica

Para representar gráficamente nuestro dataset hemos cargado el xml exportado en Tableau y hemos creado algunas visualizaciones:

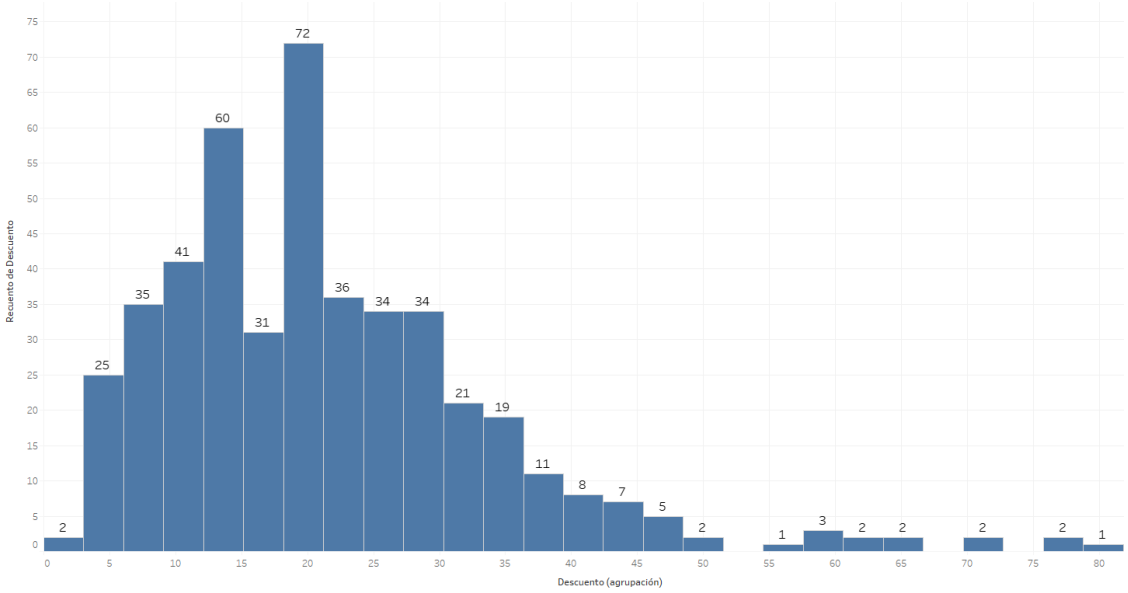
4.1. Distribución de descuentos por categorías



4.2. Distribución de descuentos por marca



4.3. Histograma descuentos



5. Contenido

Los datos se han extraído el día 31 de octubre de 2020 de la web de PCComponentes en el apartado de Outlet a través del script en Python explicado en el apartado anterior utilizando técnicas de scraping.

La estructura que tenemos es la siguiente:

- Número de fila: En este campo tenemos el número de fila sin contar las cabeceras. Empieza en 0.
- Nombre: Incluimos el nombre del producto sobre el que tenemos los datos.
- Precio: Es el precio del producto tras aplicarle el descuento.
- Precio_Original: Precio del producto antes de aplicarle el descuento.
- Descuento: Tanto por ciento de descuento que se le ha aplicado al producto al ponerlo en el outlet.
- Marca: Marca del producto en venta.
- Categoría: Indica la categoría de dicho producto dentro de la web.
- Stock: Nos indica si hay stock de este producto en ese momento en la web. Siempre será como máximo 1 puesto que las ofertas de outlet son para productos concretos puntuales.
- Identificador: Es el SKU (stock-keeping unit) o identificar único del producto dentro de la plataforma.

6. Agradecimientos

En este apartado me gustaría agradecer a la web de PcComponentes por permitirme a través de su acuerdo legal el poder leer estos datos.

7. Inspiración

Como he comentado en la introducción soy un apasionado de la tecnología y llevo años realizando compras en este portal web que ha crecido mucho en los últimos años. Siempre ofrecen precios competitivos y, en concreto, la sección de outlet es un lugar donde puedes encontrar auténticas gangas.

A través de la explotación de este data set, si se conservara un histórico, se podría llegar a estimar qué marcas o categorías llegan a tener más productos en oferta o las ofertas más interesantes de forma que se pueda ahorrar más dinero aún.

8. Licencia

9. Código

Todo el código utilizado está subido en el siguiente repositorio de GitHub de libre acceso:

https://github.com/DavidPoggio/ScrapingOutletPC

10. Dataset

Al igual que el código, podemos encontrar los resultados obtenidos en el repositorio indicado anteriormente.