2021-11-10

Anton-Bravo, Adolfo

Contents

- State "PDF" from *2021-11-11 Thu 07:19*

1 Contenidos

- Hemos trabajado con algunas herramientas de recolección de datos y las seguiremos utilizando diariamente a ser posible.

- Para analizar los datos también hemos abordado algunas herramientas de línea de comandos.

- En esta ocasión vamos a trabajar con OpenRefine.

- Para ello hay que descargarse `https://openrefine.org/`

- Vamos a trabajar con datos recopilados por el proyecto TRESCA.

1.1 OpenRefine

- Nació como un proyecto de Google de software libre, Google Refine.

- Cuando Google decidió dejarlo, la comunidad de usuarixs y desarrolladorxs hicieron un "fork" del proyecto y comenzó Open Refine, que es la evolución "natural" de Google Refine.

- Por eso hay quien se refiere a el software como Google Refine, Open Refine o, directamente, Refine.

1.1.1 Los datos

- El archivo `feliz.csv` proviene de los datos recopilados por TRESCA para explorar las tendencias de Twitter durante un periodo de la pandemia del COVID19.

- El nombre responde a que de todo ese bruto de datos recopilados se han seleccionado los mensajes que contuvieran la expresión "feliz". (aunque esto tiene truco, lo recordamos en la siguiente sesión).

- Se recopila a través de la API de Twitter.

- Conviene señalar que Twitter devuelve por "tendencias" tanto los "hashtags" más usados (es decir, aquellas expresiones que comienzan

por una almohadilla #, como por ejemplo `#FelizNavidad`) como las expresiones que detecta que se usan más aunque no estén marcadas por lxs usuarixs como hashtag (como por ejemplo `Feliz Navidad`.

- Hay dos campos de fecha.

- Un campo del término de la tendencia y otro de la consulta de la tendencia.

1.1.2 Funciones

- Sirve para limpiar y analizar datos.

- Hay que tener en cuenta que la visualización de datos también se refiere a esta etapa en la que el análisis de ciertas cantidades de datos se va a apoyar en herramientas visuales que favorecerán ese análisis.

- Se le considera la "navaja suiza" del periodismo de datos por todo lo que se puede hacer "limpiando datos".

1.1.3 Instalación

- Se ha procedido a la instalación contemplando los diversos escenarios: máquinas W$ con Java, sin Java, OSX o GNU/Linux.

- También se han superado las advertencias de W$ u OSX sobre instalar software "no verificado".

  - Un paréntesis sobre esto. Por norma general, esos avisos son una buena costumbre y previenen de instalar cosas que no estén verificadas/gestionadas/testadas por quienes hacen el sistema operativo.

- La contraparte es que mucho software libre, que por su naturaleza es de código abierto, que cualquier persona puede investigar para ver si es o no dañino, que suele tener una comunidad de desarrolladorxs y usuarixs que lo "quieren" y que puede no pasar por los procesos de esas compañías de sistemas operativos propietarios también se ven incluidos en el mismo "saco" que los anteriores.

- Aunque no es un dogma de fe, si el software es libre sueles poder fiarte más que si no lo es, pero depende de ti o de que confíes en quien te lo ofrece o quien te lo recomienda que sigas ese consejo o no. Lo importante aquí es que cualquiera tiene, potencialmente, la capacidad para investigar si efectivamente es bueno o no.

1.1.4 Ejecutar Refine

- En Windows y Mac hay que pinchar sobre el ejecutable mientras que en GNU/Linux tenemos que ir desde la terminal al directorio donde se encuentra el ejecutable y ejecutarlo con `./refine`.

- Se abre una pantalla negra que no hay que cerrar, es la del programa.

- En el navegador se debería abrir una pestaña nueva (o el navegador con esa página si no lo teníamos abierto) donde aparece la interfaz de Refine.

- Refine es una aplicación web **PERO** no requiere de conexión a Internet, se ejecuta localmente (esa es la pantalla negra).

- Como toda aplicación web cliente-servidor, monta un servidor en el puerto `3333`, que es donde nos conectamos para verlo.

- El nombre de nuestro ordenador es `localhost` y su dirección IP (reservada) es `127.0.0.1`.

- Recordad que se puede ver esto en el archivo `/etc/hosts`

- Podemos tener varias pestañas abiertas pero todas serán de la misma instancia.

- Podemos tener varias sesiones de Refine abiertas en puertos distintos (se ve en siguiente sesión de Refine).

1.1.5 Primeros pasos

Se han visto algunas acciones y se repasarán en la siguiente sesión.

1. Cargar archivos

2. Editar columnas

3. Transformaciones comunes

4. Ordenar columnas

5. Facetas (numéricas, temporales y de texto)

6. Borrar filas

7. Exportar proyecto

Se recomienda seguir probando libremente ya que no se produce una pérdida de los datos pues el csv original sigue intacto (siempre que no lo borréis).

1.1.6 Manuales

- La propia página de Open Refine.

- Manual de Open Refine: `https://flowsta.github.io/refine`

1.2 Virtualización y/o alternativas a Cygwin o la terminal de OSX

El orden no importa en este caso pero, en los casos de problemas con Windows, se puede optar por probar estas alternativas:

1.2.1 Resolver los problemas de Cygwin

- Encontrar problemas no es malo, podemos aprender mucho de su resolución.

- En lo que respecta a git se recomienda seguir el siguiente procedimiento:

  - Quitar `git` con `apt-cyg remove git`.

  - Instalar `gcc-core`, `libcurl4`, `libcurl-devel`, `openssl`. Recordad que se pueden buscar estos paquetes con `apt-cyg searchall lo-que-queramos-buscar`.

  - Instalar `git` con `apt-cyg install git`

1.2.2 WSL

- Es una terminal de un sistema operativo GNU/Linux virtualizado por M$ para W10.

- Había que iniciar una PowerShell con permisos de administración y ejecutar `wsl -install` e instala una terminal de Ubuntu GNU/Linux.

- Si en vez de Ubuntu se quiere otra distribución, se puede especificar con `wsl -install -d NombreDistribución`, donde una puede ser "Debian"

1.2.3 Git bash

- En la página oficial de git, `https://git-scm.org`, se puede descargar una terminal para trabajar con git.

- Se trata de una terminal basada en Msys2 (ver más abajo).

- Viene con git y otros comandos preinstalados.

- Si se quiere utilizar como terminal multipropósito, se recomienda aprender de MSYS2.

1.2.4 Msys2

- En la página lo explican muy bien: `https://www.msys2.org/`

- Es otro proyecto de terminal multipropósito llamada Mintty que tiene cosas de Cygwin pero es independiente.

- A efectos de uso, una diferencia fundamental es la gestión de paquetes que no se realiza con apt-cyg o el software de instalación (setup) sino con `pacman`, un modelo que nace en la distribución de GNU/Linux ArchLinux: `https://archlinux.org/pacman/`

- Para aprender un poco de cómo gestionar (buscar, instalar, actualizar, borrar) paquetes se puede ir a `https://www.msys2.org/docs/package-management/` o `https://www.lifewire.com/using-the-pacman-package-manager-4`

1.2.5 Github CLI

- Github también provee una terminal CLI para relacionarse con Github: `https://cli.github.com/`

- Tiene algunos puntos a favor pero también otros en contra.

- A favor:

  - Su aspecto está muy pulido.

  - Hacen fácil y más intuitivas las acciones que se realizan.

- En contra:

  - Solo sirve para Github, no podríamos usarla con otros repositorios git que no fueran de Github.

  - Tiene una personalización para esa facilidad que hacen que sus acciones no sirvan en otras plataformas o programas.

1.2.6 VirtualBox

- No tendría por qué ser la última opción a abordar pero, en el caso concreto de tener el problema con git, sí.

- Virtualbox es un programa libre de Oracle disponible para W$, OSX o GX que sirve para virtualizar sistemas operativos.

- Es decir, desde un ordenador W$ se puede instalar en ese programa otro sistema operativo, ya sea otra versión de W$, de OSX o de GX, y esto funciona como un programa.

- De esta forma se pueden probar otros sistemas operativos o alguna de sus aplicaciones sin tocar el sistema operativo de nuestro ordenador.

- Se puede descargar de `https://www.virtualbox.org/`

1.3 Cómo poner una imagen en Github

Ha habido varias dudas sobre cómo poner una imagen en Github. Esto lo he comentado en clase el truco pero lo pondré más detallado por aquí:

1. La opción fácil es mirar cómo lo he hecho yo en uno de los documentos markdown que os he compartido y hacer lo mismo. No sé si alguien ha hecho esto, no tengo constancia.

2. Otra opción muy fácil que habéis empleado es el "modo Windows", arrastrar al editor de Github y que ponga la ruta. Eso, si bien no es lo que se pide, también puede servir para que nos dé pistas de cómo hacerlo.

3. Hacerlo bien.

Para hacerlo bien tenemos que tener en nuestro repositorio git la misma imagen que queremos tener en Github, siempre que esta imagen sea "nuestra", es decir:

- Que tenga la propiedad de la imagen, porque la he tomado o creado con software. En este caso no hace falta especificar que se trata de una imagen tuya, pues todo el contenido tuyo es tuyo, pero sí que conviene ponerlo.

- Que la haya generado yo, por ejemplo, a través de una captura de pantalla de una infografía de una página web. En este caso conviene poner un aviso de "Captura de pantalla realizada de la web-que-sea cuya propiedad intelectual pertenece a quien-sea".

Entonces, si la imagen es "nuestra":

- En la carpeta del repositorio, creamos una carpeta para las imágenes de nombre, por ejemplo, `img` con `mkdir img`.
- Copiamos la imagen con `cp ruta-imagen-origen img/`.
- Editamos con `nano` nuestro archivo y enlazamos la imagen con `![texto de la imagen](/img/nombre-de-la-imagen "title o texto que aparece cuando pasamos por encima")`. Guardamos y salimos.
- Nota: para crear un enlace de texto es igual, solo cambia que no se coloca el signo de cierre de admiración delante.
- Miramos el estado de nuestro repositorio con `git status`. Nos dirá que hay archivos sin seguimiento.
- Los añadimos con `git add ruta-archivos`
- Comentamos la acción con `git commit -m "comentario útil"`
- Lo subimos con `git push`

1.4 Problemas Ruby

Habría que probar esta posible solución `https://gist.github.com/ahsankhatri/260c58929962a01bf9e26ef660c832d0`

- Seleccionar los paquetes curl, git, nano, openssh, openssl, unzip, util-linux, vim, wget
- Instalar ruby desde su página `https://rubyinstaller.org/downloads/`

- Cambiar la variable PATH para tener a Ruby en el PATH `/cygdrive/c/Ruby22/bin` (o algo parecido, hay que comprobarlo)

- Editar `~/.bash_profile` o `~/.profile` con `alias gem=gem.bat`

- Instalar lolcat con `gem install lolcat`.

- A partir de ahora, cuando instalas un paquete (gema) hay que añadir a `~/.bash_profile` esto: `alias lolcat=lolcat.bat`

2 Pruebas

- Describe los datos que estamos utilizando en el proyecto TRESCA. Qué tipo de archivo y qué tipo de datos.

- Explica la diferencia entre filas y columnas

- ¿Cómo harías que OpenRefine interpretara correctamente los tipos de datos?