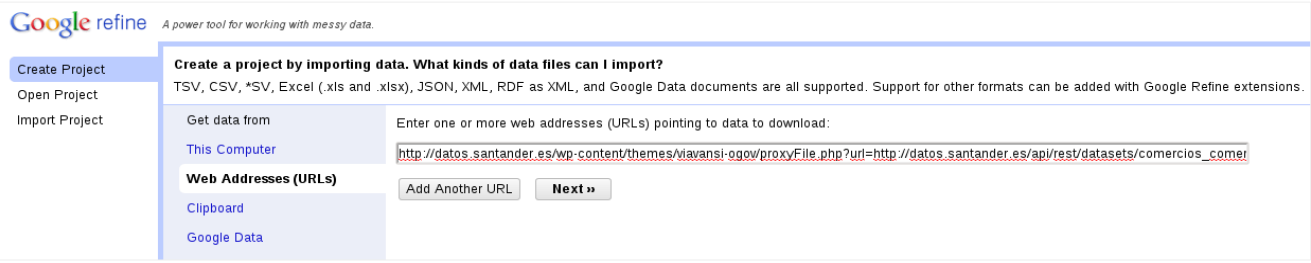
Normalización de datos con openRefine

Todos nos hemos tenido que enfrentar en algún momento con junglas de datos provenientes de fuentes donde parece que la normalización es para cobardes. Datos destipificados o incosistentes debido a errores humanos al mecanizarlos o sencillamente a diseños para el almacenamiento de datos pobremente trabajados. Esta falta de homogeneización nos puede causar no pocos quebraderos de cabeza cuando se trata de realizar un análisis de la información, así que debemos abordar su normalización antes de comenzar cualquier proceso.

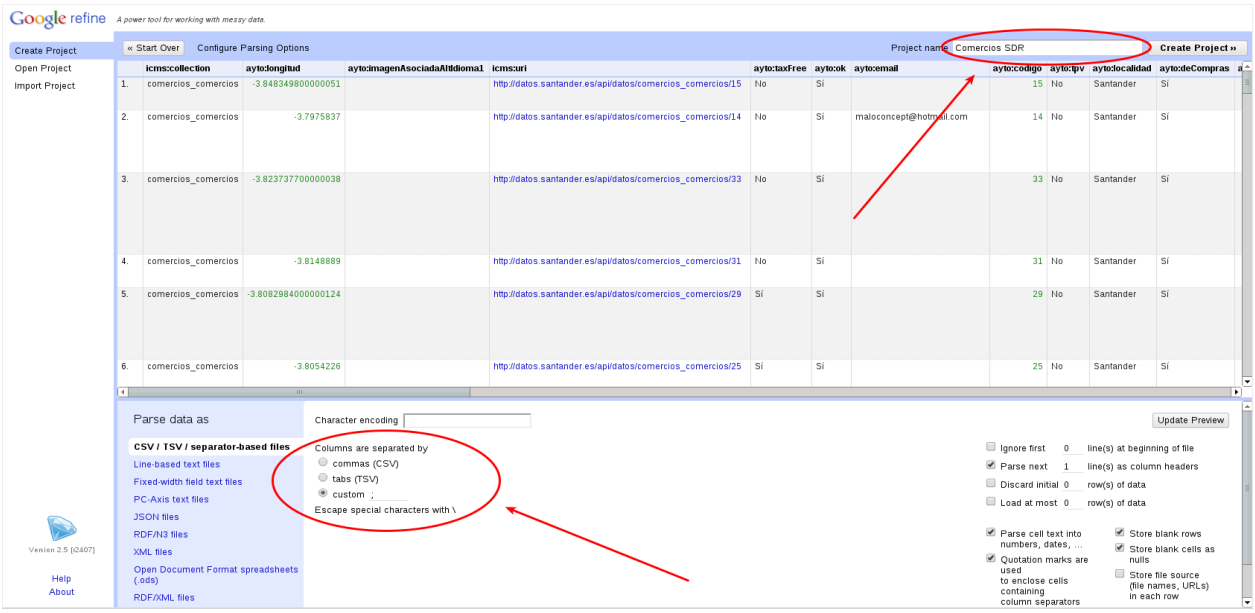
Si te gusta vivir al limite quizá el primer arrebato sea lanzarte a organizar el desaguisado, pero pronto te darás cuenta que aquello se convierte en un círculo vicioso que se repite una y otra vez, ad infinitum. En esta entrada nos hemos propuesto utilizar OpenRefine para ello, una aplicación de código abierto especialmente dirigida para actuar sobre estos datos desorganizados, facilitando su limpieza y depuración para luego poder trabajar con ellos en otros programas informáticos y continuar nuestro flujo de trabajo.
No existen demasiadas herramientas open source en el campo de la limpieza de datos, quizá las más conocidas sean Open Refine y DQGuru. OpenRefine (anteriormente conocido como Google Refine) es una aplicación web, por lo tanto vamos a trabajar de manera offline directamente en nuestro navegador. No requiere enviar ni recibir ningún dato a través de Internet. Una vez descargado y descomprimido el archivo de la web del proyecto lo ejecutaremos.
La aplicación arrancará mediante un servidor web local, iniciándose en el navegador en el que se abrirá la interfaz web de usuario.

Inicialmente podemos seleccionar el tipo de importación desde diferentes tipo de fuentes de datos. En nuestro caso para este ejemplo utilizaremos los datos facilitamos por el portal Santander Datos Abiertos, una de las iniciativas de la línea estratégica correspondiente a Open Data recogida en el Plan Santander Smart City. Trabajaremos con el juego de datos de comercios al por menor situados en el municipio de Santander. Del conjunto de formatos disponibles seleccionaremos el CSV.

Aunque como he dicho trabajaremos de manera offline, OpenRefine nos da la opción de descargarnos los datos desde una fuente remota directamente desde Internet, y es lo que vamos hacer. Copiaremos la URL que nos facilitan y en OpenRefine seleccionaremos **Web Addressess (URLs)** y pegaremos directamente la dirección web que apunta a los datos para su descarga:
http://datos.santander.es/api/rest/datasets/comercios_comercios.text?items=2011&rnd=1455092917

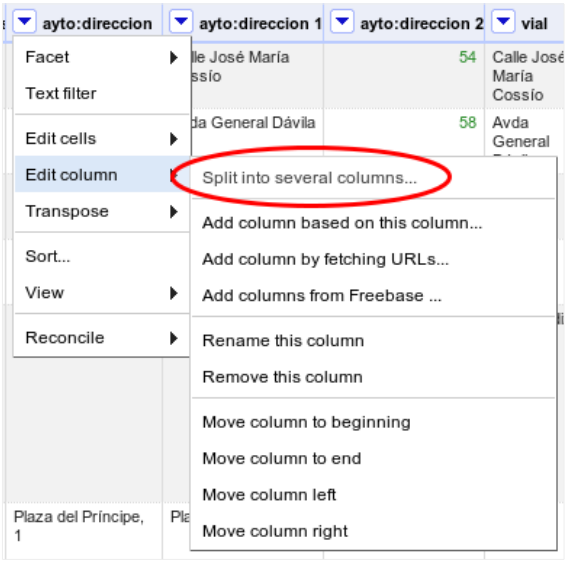


OpenRefine analizará los datos y tratará de detectar automáticamente la configuración correcta de importación. Si todo sale bien veremos una ventana de previsualización de datos. Nos aseguraremos que el tipo de datos sea CSV, la codificación correcta de los caracteres, daremos un nombre al proyecto y pulsaremos **Create Project**.

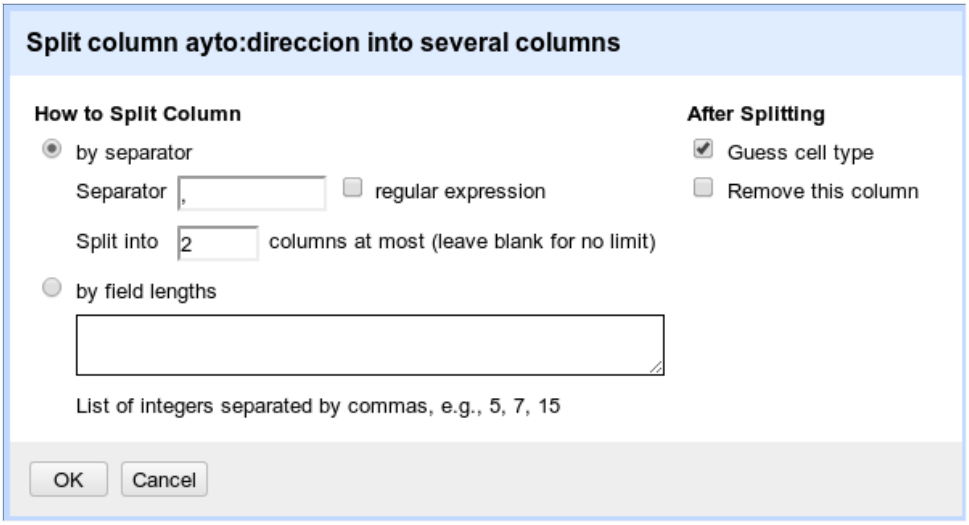


Como podemos ver OpenRefine no difiere mucho estéticamente de programas de hojas de cálculo. En este ejemplo vamos a trabajar intentando normalizar las direcciones locales de los establecimientos recogidos en la columna **ayto:direccion**, desglosando estos datos en columnas. Esta es una de las tareas previas básica a la hora de iniciar un proceso de geocodificación en cualquier Sistema de Información Geográfica (SIG).
Observamos que el número de policía suele estar separado del nombre de la calle por una coma. Esto nos facilitará mucho el trabajo, pero no siempre será así. Hay que tener muy presente que el refinado de datos suele ser un proceso iterativo: hacemos un par de transformaciones, vemos cómo ha ido, y lo intentamos mejorar.

Para ello los primero que haremos será desplegar el submenú de operaciones pulsando el botón de la cabecera de la columna, y para agilizar el proceso seleccionamos **Edit column -> Split into several columns**.



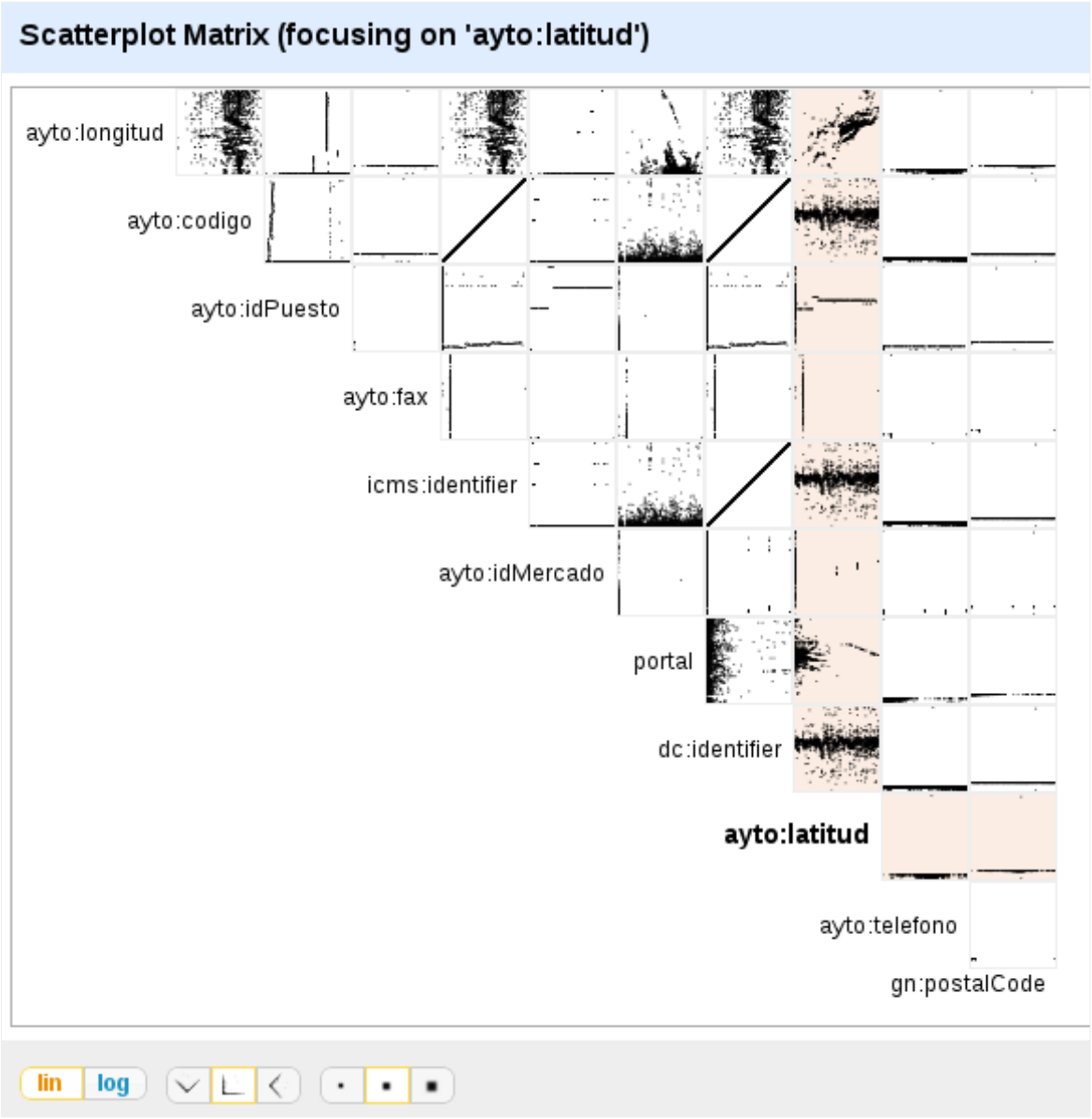
Mantenemos como caracter separador la coma, indicaremos que serán dos el número máximo de columnas que se crearán y desmarcamos **Remove this column** para que mantener los datos de origen.



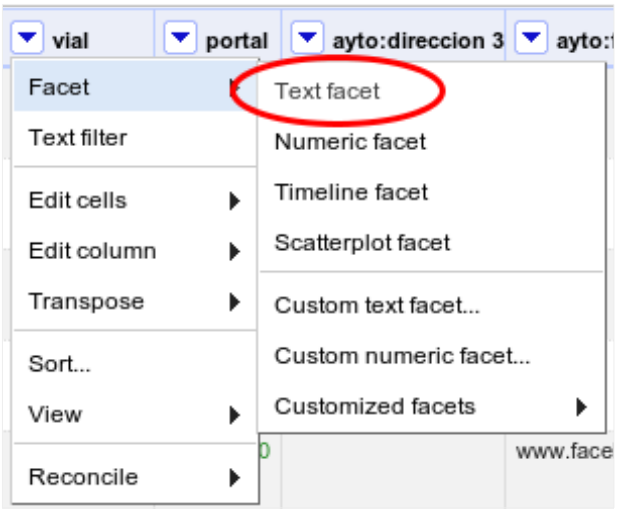
Vemos como se han creado varias columnas de una sola vez al dividir la columna original seleccionada. Renombraremos las columnas “ayto:direccion 1″ com “vial” y “ayto:direccion 2″ como “portal” desde el submenú de funciones **Edit column -> Rename this column**.
Hasta aquí ningún problema. Nada que no se pueda hacer con un poco de maña de manera más o menos sencilla mediante una hoja de cálculo. Estas son buenas para números pero no tanto para cadenas de caracteres, que es donde reside la fortaleza de OpenRefine.

El siguiente paso es normalizar la gran cantidad de ligeras variaciones que existen para un mismo nombre de

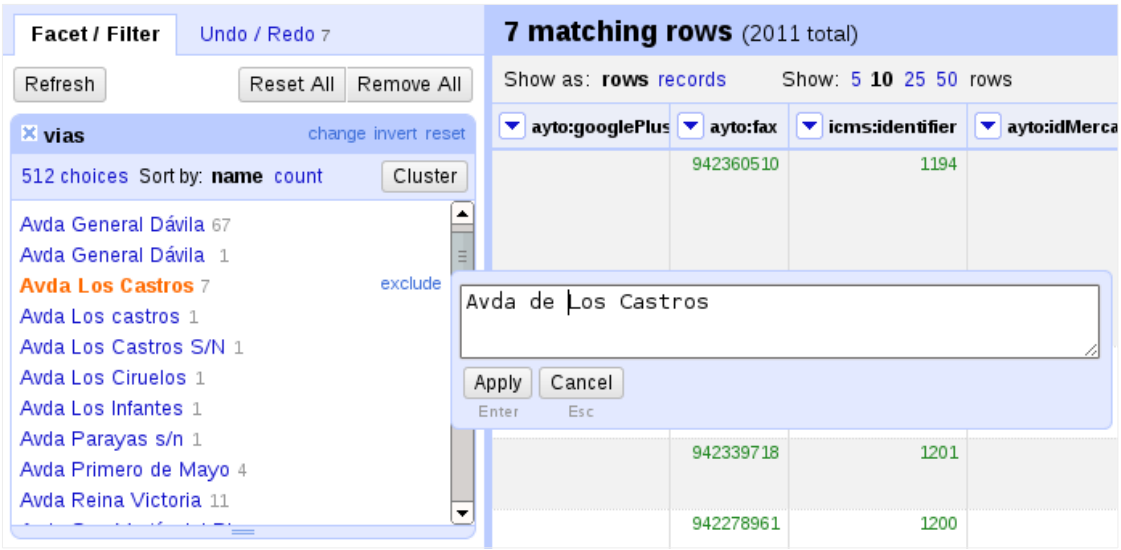
calle. Necesitamos por tanto combinar y estandarizar estas redundancias. Para ello utilizaremos las llamadas facetas, que en OpenRefine son filtros de análisis sintácticos que comparan los datos de una columna dada y nos indicará los valores únicos. Esta función es una de las principales herramientas de OpenRefine para limpieza de datos semidesasistida y nos facilita información, como los gráficos de dispersión, para localizar de un vistazo patrones numéricos divergentes.



Volvemos a abrir el submenú de operaciones haciendo clic en el encabezado de la columna “vial” recién creada y seleccionamos **Facet -> Text Facet**.



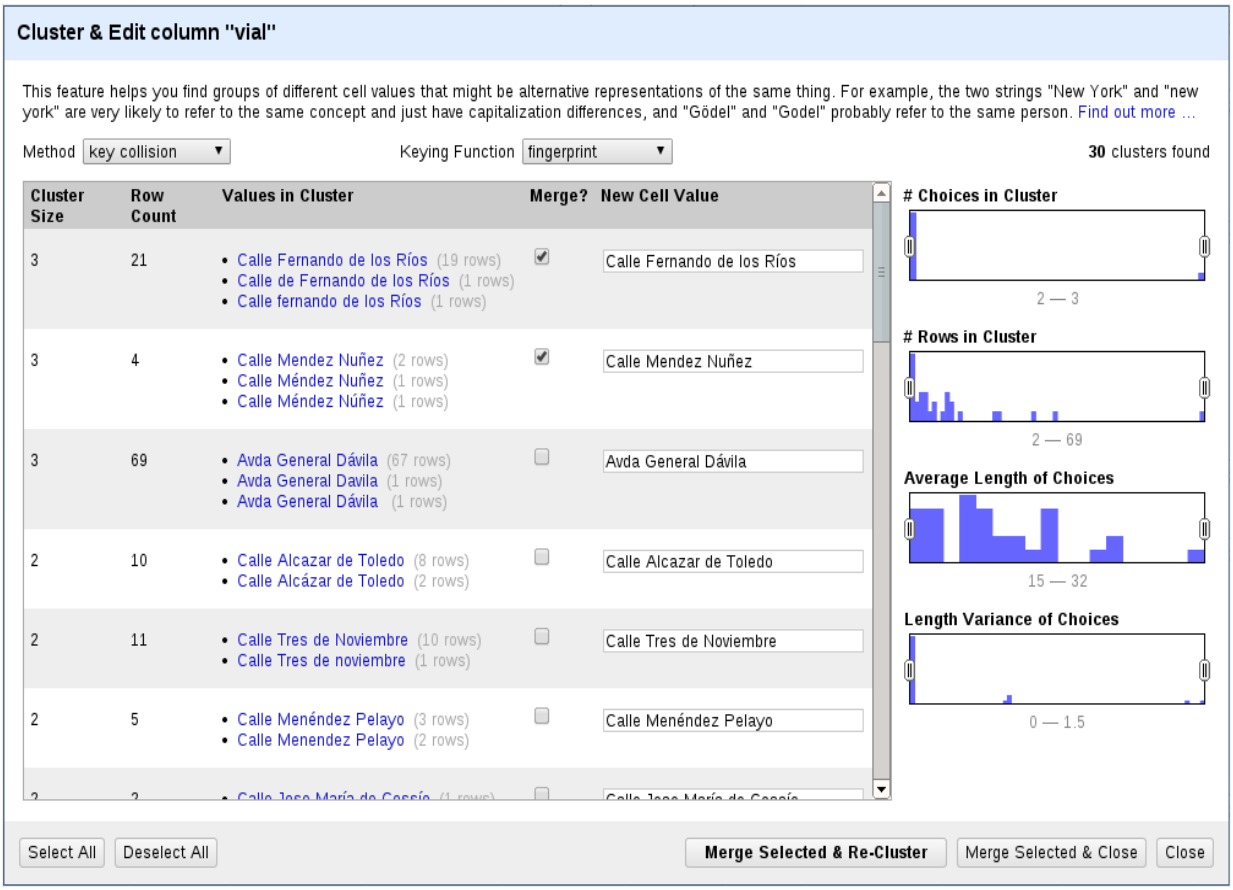
Si echamos una vistazo al panel de la izquierda OpenRefine ha listado los nombres de las calles únicas con el número de registros que coinciden con cada una de ellas. Como vemos en nuestro ejemplo existen diferentes variaciones para “Avenida de Los Castros” entre otras. Seleccionando las variaciones incorrectas y pulsando **Edit** podremos ir corrigiendo por lotes estas alteraciones.



Si pulsamos sobre el botón **Cluster** del mismo panel, el buscador facetado encuentra los grupos de diferentes valores de las celdas que podrían ser representaciones alternativas de la misma cosa, permitiendo que los datos se dividan en características manejables para el análisis. El proceso tiene opciones de personalización y varios algoritmos de búsqueda, pero pueden requerir una gran cantidad de memoria si el el número de datos es grande:

- **Fingerprint** es la función más rápida y simple y funciona relativamente bien, por lo que viene activada por defecto. Normaliza las cadenas de textos a minúsculas, las transforma a caracteres ASCII, elimina espacios y símbolos de puntuación, etc. Es útil para problemas en el orden de las palabras. (Corrales de Buelna, Los -> Los Corrales de Buelna)
- **Ngram** se utiliza para las transposiciones de caracteres (Brugos -> Burgos; Paísvasco -> País Vasco).
- **Metaphone3** y **Cologne phonetic** son funciones de comparación fonética que, aunque orientadas a la

pronunciación en idioma inglés y el alemán respectivamente, pueden ser de utilidad.



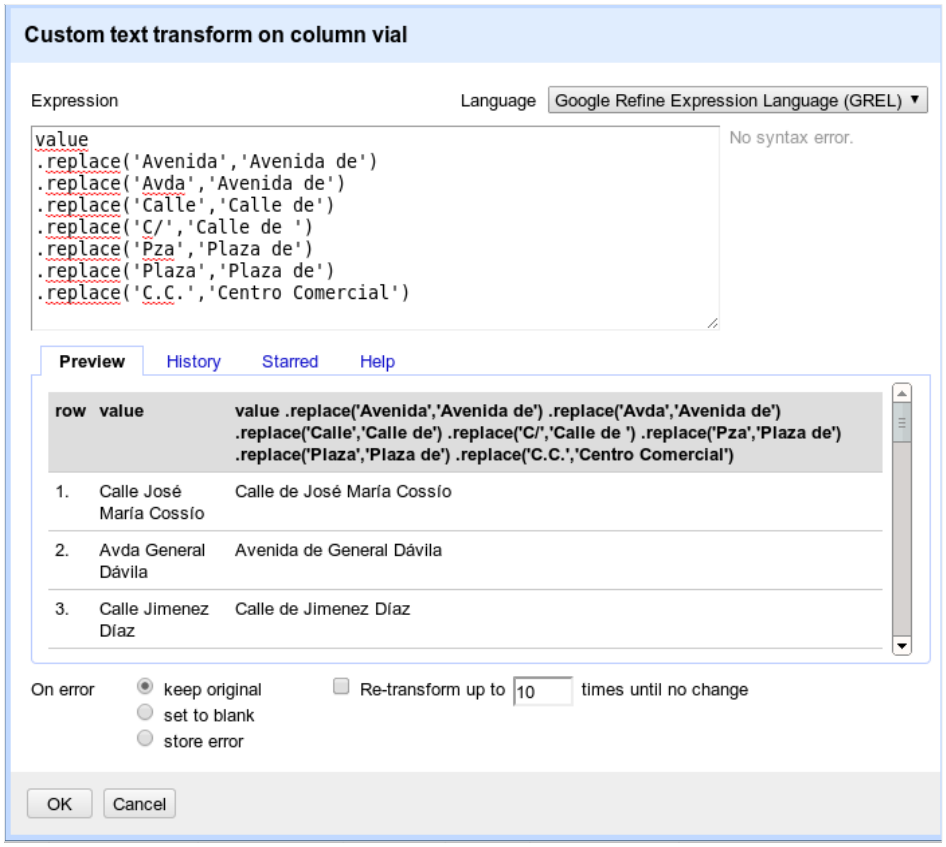
Combinaremos los grupos correctos marcando las casillas de la columna **Merge** y pulsaremos **Merge Selected & Re-cluster**. Repetiremos el proceso tanteando de nuevo los métodos y funciones existentes para ir reduciendo paulatinamente los clústeres de datos divergentes.

OpenRefine nos permite crear nuestras propias facetas ad hoc usando GREL, Jython o Clojure. GREL(Google Refine Expression Language) es un lenguaje de programación con un gran número de funciones que permite realizar tareas de depuración avanzadas. Veamos algún caso sencillo.

Vamos a modificar las abreviaturas de los nombre de los viales a su denominación extendida. En el submenú de operaciones de la columna “vial” seleccionaremos **Edits cells -> Transform** e introduciremos la siguiente expresión GREL:

```
value.replace('Avda','Avenida')
```

Las expresiones las podemos encadenar complicándolas tanto como queramos. El ejemplo de la imagen siguiente realiza múltiples reemplazos de una sola vez.



Observa como en algunas de celdas existen direcciones donde se especifica más detalladamente la localización del establecimiento entre paréntesis, por ejemplo Calle Cádiz 4 (Plaza de las Estaciones). Por cualquier razón no queremos perder esos datos y deseamos almacenarlos en una nueva columna. Para ello en la columna “portal” seleccionaremos **Edit column -> Add column based on this columns…**, llamamos a la nueva columna “detalle” e introducimos la siguiente expresión:

```
value.split('(')[1].replace(')','')
```

Esta consta de las funciones split y replace. Con la primera extraemos todo aquello que queda tras el signo de paréntesis izquierdo y con la segunda eliminamos el signo de parántesis derecho del resultado. A continuación vamos a las celdas de la columna “portal” para eliminar el contenido entre paréntesis que ya no necesitamos. Para ellos accedemos al menú **Edits cells -> Transform** e introducimos el siguiente código:

```
value.replace(/\(.*/,'')
```

La expresión GREL eliminará cualquier caracter alfanumérico desde el símbolo de paréntesis abierto. Hay que señalar por último que OpenRefine mantiene un registro de todo el trabajo realizado en un proyecto. Este registro se puede utilizar para deshacer y rehacer las transformaciones o para ser guardado como JSON para su uso con otros conjuntos de datos. Esto hace que sea muy apropiado para trabajar con grupos de archivos similares que requieren tratamientos análogos.

En una próxima entrada en este blog explicaré como enriquecer estos datos. La idea es ofrecer un mayor valor añadido a los datos contextualizándolos mediante las geocodificación de las direcciones postales.