

## Práctica 1: Limpieza con OpenRefine.

<b>Objetivo general</b>	<b>2</b>
<b>Descripción</b>	<b>2</b>
<b>Problemas</b>	<b>2</b>
Registros en blanco	2
Columna sin utilidad	4
Categorías mal escritas	4
Redundancia en columnas	8
Datos descolgados	9
Datos sin utilidad	10
<b>Formato JSON</b>	<b>12</b>
<b>Metodología</b>	<b>13</b>
<b>Mejoras posibles</b>	<b>13</b>

## Objetivo general

Con esta actividad vas a llevar a cabo una de las tareas más importantes y comunes para un científico de datos: la limpieza de datos. Revisarás cada variable, comprobarás la existencia de errores, identificarás datos ausentes, realizarás conversión de formatos si aplica y finalmente, una vez preparado el dataset, lo llevarás a una base de datos.

## Descripción

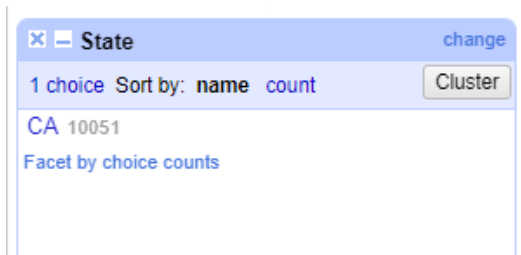
Los datos para esta actividad están en el fichero data\_act\_01.csv. Sobre dicho fichero llevarás a cabo la limpieza de datos. Existen al menos (6) seis errores que deben ser identificados y corregidos, con el fin de limpiar y homogeneizar el dataset final.

## Problemas

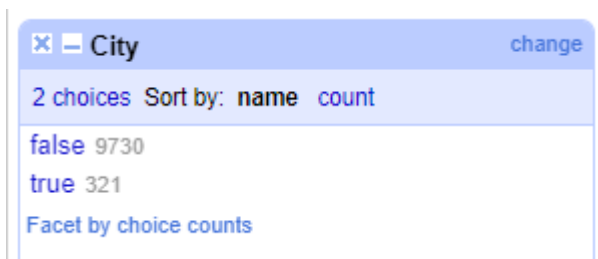
### Registros en blanco

El **primer problema** que he detectado es que en la columna State, de los 10.051 valores que tenemos en el csv, 10.048 contienen el valor “CA”, pero hay 3 en blanco. Esto lo he podido detectar utilizando un facet personalizado para registros en nulo o strings vacíos. Viendo estos números, interpreto que las 3 filas vacías son errores y que su valor realmente es “CA”, como todo el resto de filas. Por lo tanto, mi solución ha sido incluir el valor “CA” en las tres filas.





El **segundo problema** identificado es que hay otra columna que contiene registros en blanco y es la columna “City”, en este caso hay un problema mayor, puesto que no se trata de 3 filas, sino de 321, lo cual representa un 3% de los datos. Esto lo he podido detectar de igual manera que en el caso anterior:



En este caso, una posible solución sería eliminar esas 321 filas del documento para solo tener datos completos, puesto que representan solamente un 3%, pero he decidido no hacerlo y optar por dejar estas filas sin ciudad. Esto lo he hecho por dos razones, la primera, que el csv cuenta con otra columna llamada Address, la cual no tiene ningún dato en blanco, por lo que podría llegar a conseguirse la ciudad en un tratamiento futuro, y la segunda y es la que más convincente me ha parecido, es que en el caso de estas 321 filas, en la columna AddressType, el valor es “Geo-override” en 318. He buscado el significado de este concepto, ya que lo desconocía y he preguntado a la IA dando contexto:

En algunos sistemas o aplicaciones, cuando no se dispone de datos de ubicación específicos o cuando se desea proporcionar una ubicación genérica para registros sin información de ubicación precisa, se puede utilizar el valor "geo override" en la columna "AddressType" para indicar que se ha sobrescrito la geolocalización predeterminada con un valor personalizado o genérico.

Por ejemplo, en una aplicación de mapeo, si un registro no tiene información de ciudad específica pero se conoce su ubicación general, podría asignarse el valor "geo override" en la columna "AddressType" para indicar que se ha sobrescrito la ubicación predeterminada con una ubicación genérica.

En resumen, la presencia del valor "geo override" en la columna "AddressType" en los registros donde la columna "City" está vacía podría indicar que se ha realizado una sobrescritura o anulación de la geolocalización predeterminada para esos registros específicos, proporcionando así una ubicación genérica o personalizada en lugar de datos específicos de la ciudad.

Sabiendo esto, he interpretado que el hecho de dejar la columna “City” vacía podría ser algo intencionado o ya controlado por la persona que ha introducido los datos, el valor “Geo-Override” se encuentra en más registros que sí cuentan con un valor en “City”, pero viendo que en 318/321 que tienen este valor vacío se cumple y sabiendo el significado, me ha parecido que podría estar utilizándose a modo de justificación, y teniendo en cuenta también el primer punto del que hablaba, he decidido borrar los 3 valores de esos 321 que no tienen ciudad y que no tienen el valor “Geo-Override” en vez de los 321 completos, ya que en estos sí que considero que es un error.

<div> <div>AddressType</div> <div>change</div> </div> <div> 3 choices Sort by: name count <div>Cluster</div> </div> <div> 1 2 </div> <div> Common Location 1 </div> <div> Geo-Override 318 </div> <div> Facet by choice counts </div>	<div> <div>City</div> <div>change invert reset</div> </div> <div> 2 choices Sort by: name count </div> <div> false 9730 </div> <div> true 318 <div>exclude</div> </div> <div> Facet by choice counts </div>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## Columna sin utilidad

El **tercer problema** que he encontrado en el documento es que hay una columna la cual no contiene ningún dato, tiene todas sus filas en blanco, lo cual la convierte en una columna totalmente inútil. Este es un error detectable a simple vista, aunque para asegurarme he utilizado una faceta contra valores blancos o strings vacíos:

<div> <div>Range</div> <div>change</div> </div> <div> 1 choice Sort by: name count </div> <div> true 10051 </div> <div> Facet by choice counts </div>
-------------------------------------------------------------------------------------------------------------------------------------------------------

Puesto que no tiene ningún dato y no está dando ningún tipo de información al documento, he decidido que la solución era borrarla.

## Categorías mal escritas

El **cuarto problema** que he podido detectar ha sido que en la columna categórica “OriginalCrimeTypeName”, de los 575 valores registrados, hay valores que se refieren al

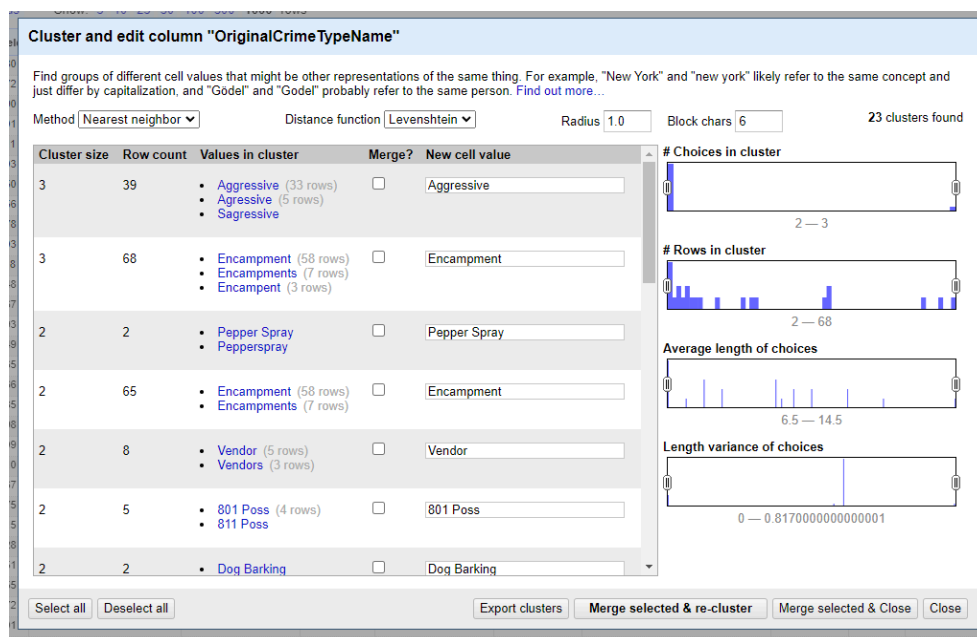
mismo tipo de crimen pero escritos de distinta forma, al parecer por error. Esto lo he identificado utilizando una faceta de texto y un cluster:

The image shows two screenshots of a data clustering tool interface, specifically the 'Cluster and edit column "OriginalCrimeTypeName"' window. The tool is used to find groups of different cell values that might be other representations of the same thing.

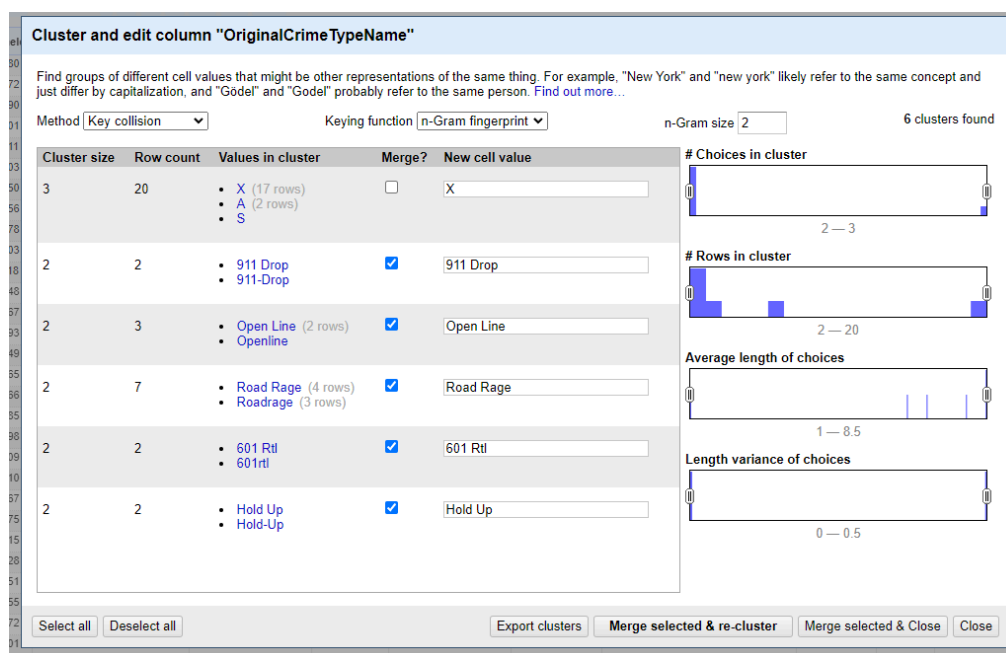
**Top Screenshot:** The interface shows a table with columns: Cluster size, Row count, Values in cluster, Merge?, and New cell value. The 'Method' is set to 'Key collision' and the 'Keying function' is 'Fingerprint'. The table lists several clusters, including 'R/o Violation', '800 Poss', '909', 'H&r', 'Susp Poss', 'lp', and '519 Poss'. To the right, there are four histograms: '# Choices in cluster', '# Rows in cluster', 'Average length of choices', and 'Length variance of choices'. The histograms show the distribution of choices, rows, average length, and length variance for the clusters.

**Bottom Screenshot:** This screenshot shows the same interface after some clusters have been merged. The table now lists clusters like '519 Poss', 'Jo', 'Drugs', '801 Poss', 'Poss 5150', 'Ltd', and '152 Poss'. The histograms on the right are also updated to reflect the changes in the data.

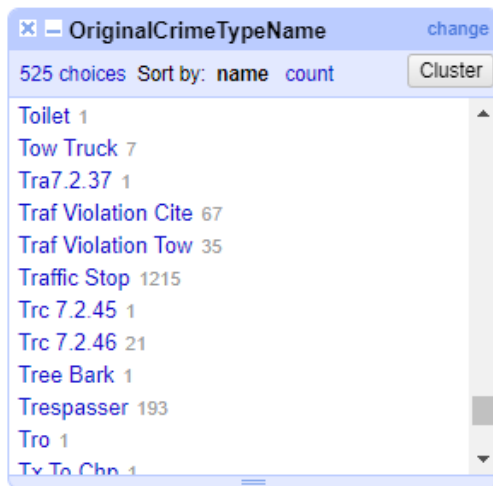
Como se puede ver, hay muchos valores que son lo mismo pero escritos cambiando el orden de las palabras, con algún símbolo... Puesto que es evidente que se refieren a la misma categoría de crimen, he decidido que la solución era hacer un merge y unificarlos en un solo valor. Una vez unificados, he realizado otra faceta y otro cluster, pero esta vez, el cluster ha sido por "Nearest Neighbor", y ha identificado incluso más errores que seguían ahí.



En este caso he tenido cuidado y solo he hecho merge en los que realmente son errores evidentes, puesto que por ejemplo, 801 poss y 811 poss, seguramente se refieran a categorías distintas. También he realizado más clusters cambiando el “Keying Function” y he podido detectar más categorías con fallos ortográficos, espacios...



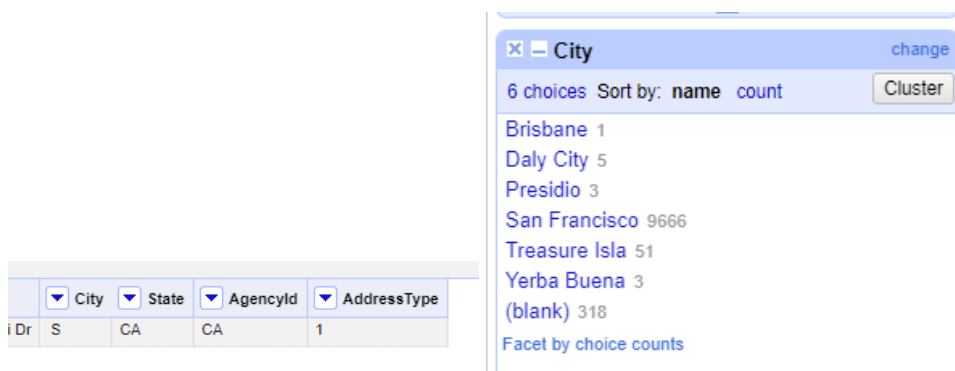
Tras unificar todos los valores que se referían a la misma categoría pero tenían algún error en su escritura, el documento ha pasado de tener 575 valores en la columna a tener 525.



El **quinto problema** que he encontrado también ha sido de valores erróneos, pero esta vez en la columna de “City”, lo he identificado con una faceta y un cluster también.



En este caso, he decidido unificar el valor de “SAN FRANCISCO” y he borrado el registro que contenía una S, ya que mirándolo, he visto que también contenía un error en la columna “AgencyID”, donde todos los valores son 1 y 2 y esta fila tiene el valor “CA”, además también tiene un valor erróneo en “AddressType”, siendo el único registro con un valor numérico en la columna, por lo que es una fila con tres errores.






El **sexto problema** y último en esta sección, lo he identificado en la columna “AddressType” de la misma manera que los dos anteriores, el error está en una de las categorías, la cual tiene un valor mal escrito:



AddressType	
5 choices	Sort by: name count
Common Location	817
Geo-Override	469
Intersection	3701
Intersectionoon	1
Premise Address	5059

La solución que he realizado ha sido unificar el valor mediante el merge del cluster.



AddressType	
4 choices	Sort by: name count
Common Location	817
Geo-Override	469
Intersection	3702
Premise Address	5059

## Redundancia en columnas

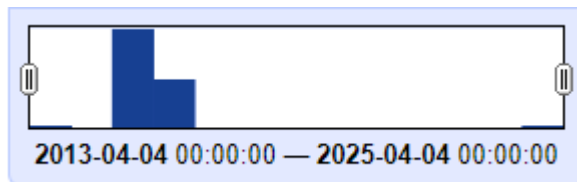
El **séptimo problema** lo he identificado en las columnas “CallTime” y “CallDateTime”, en la primera columna se nos da la información de la hora a la que se hizo la llamada por el accidente y en la segunda nos da la misma hora con dos horas añadidas y además la fecha, es decir, repite la información de la primera, lo que convierte a la primera columna en un registro inútil, ya que tenemos la misma información y más completa en la segunda.

	CallTime	CallDateTime	
DZ	18:42	2016-03-30T16:42:00Z	R
DZ	15:31	2016-03-31T13:31:00Z	G
DZ	16:49	2016-03-31T14:49:00Z	G
DZ	17:38	2016-03-31T15:38:00Z	G
DZ	17:42	2016-03-31T15:42:00Z	R
DZ	18:29	2016-03-31T16:29:00Z	G
DZ	18:43	2016-03-31T16:43:00Z	A
DZ	18:47	2016-03-31T16:47:00Z	H
DZ	18:52	2016-03-31T16:52:00Z	N
DZ	18:57	2016-03-31T16:57:00Z	A
DZ	18:59	2016-03-31T16:59:00Z	R
DZ	19:08	2016-03-31T17:08:00Z	G
DZ	19:13	2016-03-31T17:13:00Z	H
DZ	19:18	2016-03-31T17:18:00Z	G
DZ	19:37	2016-03-31T17:37:00Z	A

La solución que he visto más correcta es la de borrar la primera columna para no tener redundancia de datos, además, en caso de necesitar una columna con solo la hora en un futuro, podría solucionarse cogiendo la hora de la columna “CallDateTime”.

## Datos descolgados

El **octavo problema** que he identificado se encuentra en los datos de la columna “OffenseDate”, donde se especifica la fecha del accidente. He convertido estos datos a “time” y he hecho una faceta de línea de tiempo.



El error se encuentra en que todos los registros del documento tienen fecha en 2016, pero hay dos registros descolgados, uno en 2013 y otro en 2025.

Facet / Filter
Undo / Redo 22 / 22
Refresh
Reset all
Remove all
OffenseDate
change reset

1 matching rows (10047 total)
Show as: rows records
Show: 5 10 25 50 100 500 1000 rows

	CrimelId	OriginalCrimeTypeName	OffenseDate	CallDateTime	
7255	160950844	Traffic Stop	2013-04-03T22:00:00Z	2016-04-04T07:00:00Z	CIT



Custom text transform on column OffenseDate

Expression Language General Refine Expression Language (GREL)

`value.split("T")[0]` No syntax error.

**Preview** History Starred Help

row	value	value.split("T")[0]
1.	2016-03-29T22:00:00Z	2016-03-29
2.	2016-03-30T22:00:00Z	2016-03-30
3.	2016-03-30T22:00:00Z	2016-03-30
4.	2016-03-30T22:00:00Z	2016-03-30
5.	2016-03-30T22:00:00Z	2016-03-30
6.	2016-03-30T22:00:00Z	2016-03-30

On error ☒ keep original ☐ Re-transform up to  times until no change  
☐ set to blank  
☐ store error

OK Cancel

« first < previous

TypeName	OffenseDate	Call
	2016-03-29	2016-03-30T16:4
nt <a href="#">edit</a>	2016-03-30	2016-03-31T13:3
	2016-03-30	2016-03-31T14:4
	2016-03-30	2016-03-31T15:3
	2016-03-30	2016-03-31T15:4
	2016-03-30	2016-03-31T16:2
nt	2016-03-30	2016-03-31T16:4
nt	2016-03-30	2016-03-31T16:4
nv	2016-03-30	2016-03-31T16:5

# Formato JSON

The screenshot shows the JSONGrid interface. On the left, under the 'JSON' tab, a JSON object is displayed with line numbers 1 through 13. The object contains fields like CrimeId, OriginalCrimeTypeName, OffenseDate, CallDateTime, Disposition, Address, City, State, AgencyId, and AddressType. On the right, under the 'GRID' tab, the same data is presented as a table with 2 columns: the field name and its value.

CrimeId	1
OriginalCrimeTypeName	Susp Info
OffenseDate	2016-03-24
CallDateTime	2016-06-04T00:00:00Z
Disposition	GOA
Address	100 Block Of Chilton Av
City	San Francisco
State	CA
AgencyId	1
AddressType	Premise Address

En el posible formato JSON que he aportado ya están eliminadas las dos columnas “Range” y “CallTime” descritas en el apartado anterior. También está cambiado el formato de “OffenseDate”, donde no se especifica la hora y en el caso de los valores numéricos no tienen comillas para respetar la sintaxis y especificar que solamente pueden ser numéricos. Para pasar el contenido del csv a json he utilizado un conversor online, el cual lo ha hecho de forma automática y respetando toda la estructura:

## CSV or TSV > JSON

To get started, upload or paste your data from Excel (saved as CSV or TSV).

Upload a CSV file

Seleccionar archivo Ninguno archivo selec.

Or paste your CSV here

```
CrimeId,OriginalCrimeTypeName,OffenseDate,CallDateTi
me,Disposition,Address,City,State,AgencyId,AddressTy
pe
160903280,Assault / Battery,2016-03-
29T00:00:00Z,2016-03-30T16:42:00Z,REP,100 Block Of
Chilton Av,San Francisco,CA,1,Premise Address
160912272,Homeless Complaint,2016-03-
30T00:00:00Z,2016-03-31T13:31:00Z,GOA,2300 Block Of
Market St,San Francisco,CA,1,Premise Address
160912590,Susp Info,2016-03-30T00:00:00Z,2016-03-
31T14:49:00Z,GOA,2300 Block Of Market St,San
Francisco,CA,1,Premise Address
160912801,Report,2016-03-30T00:00:00Z,2016-03-
31T15:38:00Z,GOA,500 Block Of 7th St,San
Francisco,CA,1,Premise Address
160912811,594,2016-03-30T00:00:00Z,2016-03-
31T15:42:00Z,REP,Beale St/bryant St,San
Francisco,CA,1,Intersection
```

Separator

Auto-detect

☒ Parse numbers

☒ Parse JSON

☐ Transpose

Output:

☒ Array

☐ Hash

☐ Minify

JSON

```
{
  "State": "CA",
  "AgencyId": 1,
  "AddressType": "Premise Address"
},
{
  "CrimeId": 160913148,
  "OriginalCrimeTypeName": "Suspicious Person",
  "OffenseDate": "2016-03-30T00:00:00Z",
  "CallDateTime": "2016-03-31T17:08:00Z",
  "Disposition": "GOA",
  "Address": "700 Block Of Eddy St",
  "City": "San Francisco",
  "State": "CA",
  "AgencyId": 1,
  "AddressType": "Premise Address"
},
{
  "CrimeId": 160913167,
  "OriginalCrimeTypeName": "Suspicious Person",
  "OffenseDate": "2016-03-30T00:00:00Z",
  "CallDateTime": "2016-03-31T17:08:00Z",
  "Disposition": "GOA",
  "Address": "700 Block Of Eddy St",
  "City": "San Francisco",
  "State": "CA",
  "AgencyId": 1,
  "AddressType": "Premise Address"
}
```

## Metodología

Crear una metodología es algo difícil, puesto que mi experiencia es corta y los documentos pueden variar mucho, pero yo seguiría un orden igual o parecido a este:

1. Identificar columnas sin uso (vacías)
2. Identificar redundancia de datos (columnas que aportan lo mismo)
3. Identificar el tipo de dato de cada columna y transformarlo si es necesario para su futuro uso (por ejemplo pasar los valores numéricos a int si están puestos como caracteres)
4. Realizar facetas personalizadas para valores nulos o cadenas vacías para identificar el nivel de completitud de los datos
5. Realizar facetas de texto, número y línea de tiempo para identificar categorías mal escritas, datos sin lógica... Y realizar la corrección
6. Clusters para identificar datos descolgados y corregirlos

Este es el proceso que he seguido inconscientemente para realizar la práctica, empezando por lo más sencillo hasta lo más difícil, también es necesario conocer el contexto del documento y sus datos, ya que sino es difícil encontrar soluciones en los momentos de duda.

## Mejoras posibles

A mi parecer, una mejora importante sería definir mejor las categorías de la columna que define el tipo de crimen, puesto que son muy liosas y hay muchos tipos muy parecidos que se podrían englobar en un solo tipo, esto ayudaría a futuros estudios e informes, ya que tal y como están ahora, pueden verse muchos datos separados en categorías distintas que realmente representan a la misma y se pueden tomar conclusiones erróneas al analizarlos.

A nivel de captura de datos, estaría bien registrar algún dato sobre la persona que ha sufrido el accidente, como la edad, sexo... Aunque sean datos muy sencillos, ayudarían también a realizar informes y conocer un poco qué características tienen las personas que sufren más robos, qué personas suelen involucrarse en peleas...

También estaría bien capturar más datos de otras ciudades, ya que los registros son solamente del estado de California, se podría añadir un poco más de variedad, puesto que de 10.050 registros, 9665 son en San Francisco y esto ayudaría a conocer los tipos de delito por ciudad, la peligrosidad en las distintas zonas del estado...