BIG DATA II
MÁSTER EN CIENCIA DE DATOS E INGENIERIA DE COMPUTADORES

JAUME CLOQUELL CAPO
jaumecloquell@correo.ugr.com

Database

Los crímenes en Chicago son un tema muy interesante para la exploración por todo tipo de razones. Una razón es la disponibilidad de grandes cantidades de conjuntos de datos sobre la delincuencia disponibles al público (de alta calidad), abiertos para que los científicos puedan minar e investigar este tipo de datos.

En este proyecto, voy a explorar más sobre el crimen en Chicago y trataré de responder algunas preguntas:

- ¿Qué crímenes son más comunes?
- ¿Que zona es más segura?
- ¿ Cuántos criminales son arrestados?
- ¿Cómo ha cambiado el crimen en Chicago a través de los años? ¿Que año ha sido el peor?

Primero, importamos los paquetes de datos científicos requeridos y obtenemos los datos. Este conjunto de datos refleja los incidentes de crimen reportados (con la excepción de los asesinatos en los que existen datos para cada víctima) que ocurrieron desde el 2001 hasta el presente. La base de datos usada puede ser consultada y descargada en catalog.

El dataset está formado por 6822518 filas y 22 columnas (ID,Case Number,Date,Block,IUCR,Primary Type,Description,Location Description,Arrest,Domestic,Beat,District,Ward,Community Area,FBI Code,X Coordinate,Y

Coordinate,Year,Updated On,Latitude,Longitude,Location) de distintos formatos y con valores NULL en algunas de ellas.

Carga de los datos

En primer lugar, es necesario eliminar la primera línea del dataset a usar ya que corresponde con el header de la base de datos. Para ello, se ha realizado este sencillo script que simplifica el proceso.

```
tail -n +2 "Crimes_-_2001_to_present.csv" > "Crimes_-_2001_to_present.tmp"
&& mv "Crimes_-_2001_to_present.tmp" "Crimes_-_2001_to_present.csv"
```

Con sólo estas pocas líneas, Pig es capaz de recorrer el fichero y cargar los datos.

```
measure = load input/Crimes_-_2001_to_present.csv' using
PigStorage(';') as (
   id,
   case_number,
   fecha,
   block,
   iucr,
   primary_type,
   description,
   location_description,
   arrest,
   domestic,
   beat,
   district,
   ward,
   community_area,
   fbi_code,
   x_coordinate,
   y_coordinate,
   year,
   updated_on,
```

```
   latitude,
   longitude
);

store measure into 'pigResults/CrimesProcessed' using
PigStorage(',');
```

Vamos a repasar con más detalle cómo lo hace:

- **load**: Indica que realice la carga de los datos del fichero Crimes_-_2001_to_present.csv.
- **using PigStorage(‘,’)**: Indica que los datos deben ser separados por el carácter delimitador ‘,’. Esto lo hará la función PigStorage utilizada cuando tenemos un conjunto de datos estructurados y delimitados por algún carácter separador. Existen otras funciones de carga y almacenamiento como BinStorage, TextStorage, JsonLoader, HbaseStorage.
- **as:** Mediante ‘as’ definimos el schema de los datos cargados del fichero para acceder posteriormente a ellos de forma más sencilla. Indicamos a continuación el nombre de cada dato recogido y su tipo. Los tipos que admite son los tipos simples de Java: int, long, float, double, chararray, bytearray, boolean, datetime, biginteger, bigdecimal. También admite los tipos complejos: tuple (conjunto de campos ordenados), bag (una colección de tuplas), y map (conjunto de datos organizados por clave/valor).
- **dump**: Usamos el operador de diagnóstico ‘dump’ para visualizar los datos recogidos en la carga anterior. Es útil para sacar los datos por pantalla.
- **explain**: El operador ‘explain’ muestra el plan de ejecución de las tareas map reduce.
- **describe**: El operador ‘describe’ saca por consola una descripción de la tupla generada. En este caso describe el tipo de datos creado llamado ‘measure’.

Experimento de datos

Como podemos ver en la descripción del dataset tenemos varias columnas que nos ayudarán a responder a nuestras preguntas. Usaremos la columna **'Year'** para explorar patrones temporales, **'primary_type'** y '**Location Description**' para investigar su relación con el tiempo.

El trabajo debe incluir:

- Una operación de proyección.
- Una operación de selección.
- Agrupamientos (group) y resúmenes de información.

Comencemos por algunas consultas genéricas, como cuántos tipos de crímenes existen?

```
A = DISTINCT(FOREACH measure GENERATE primary_type);
```

La salida es: burglary, homicide, narcotics, obscenity, ritualism, sex offense, prostitution, other offense, criminal damage, criminal trespass, weapons violation, deceptive practice, crim sexual assault, motor vehicle theft, public peace violation, other narcotic violation, non-criminal (subject specified), arson, theft, assault, battery, robbery, gambling, stalking, kidnapping, intimidation, non-criminal, non - criminal, public indecency, domestic violence, human trafficking, liquor law violation, offense involving children, interference with public officer, concealed carry license violation.

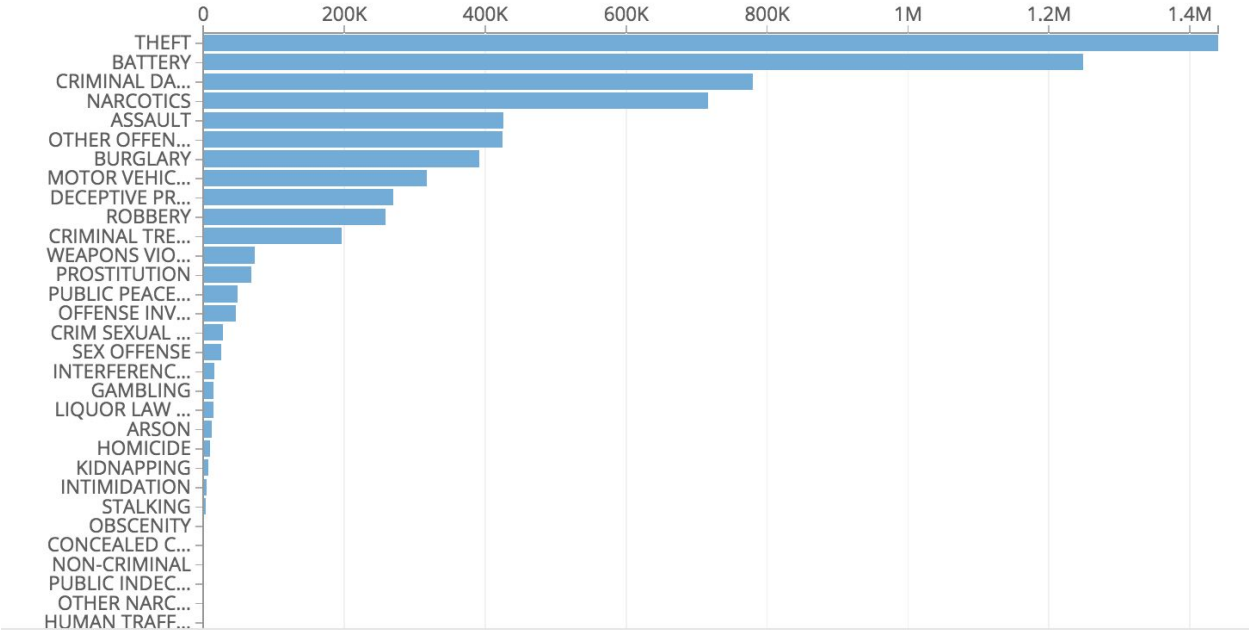
Parece que la salida ofrece más resultados de los que cabría esperar ya que existen valores que han sido introducidos erróneamente ya que a simple hay valores que por el nombre parecen ser el mismo tipo aunque hayan sido considerados distintos (**NON - CRIMINAL, NON - CRIMINAL** y **NON -CRIMINAL (SUBJECT SPECIFIED)**).

Antes de profundizar en cómo otros rasgos podrían mediar, primero quiero responder a nuestra primera pregunta ¿Qué crímenes son más comunes?

```
A = Group measure by primary_type;
B = foreach A GENERATE group, COUNT(measure) as count;
C = ORDER B BY count DESC;
dump C;
```

Tipo	Cantidad de Crímenes
THEFT	1439735
BATTERY	1248650
CRIMINAL DAMAGE	780222
NARCOTICS	716180
ASSAULT	425340

Si cuantificamos el número de crímenes por tipo y ordenamos los resultados por el cuantificador podemos obtener cuáles de los crímenes son los más comunes. Para ello, nos centraremos con los valores THEFT y BATTERY, CRIMINAL DAMAGE, NARCOTICS Y ASSAULT. Además, como se puede visualizar en la siguiente imagen, existe una gran diferencia entre el primer tipo y el tercero.



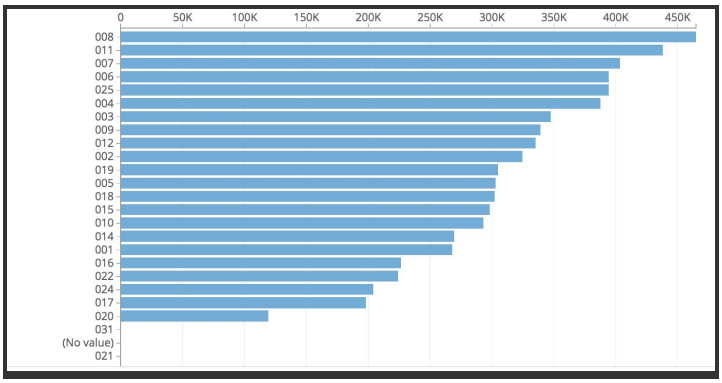
Puede ser interesante obtener el mismo resultado pero agrupadas por distrito y también localización para ver cómo varía el porcentaje de crímenes.

```
A = Group measure by district;
B = foreach A GENERATE group, COUNT(measure) as count;
C = ORDER B BY count DESC;
dump C;
```

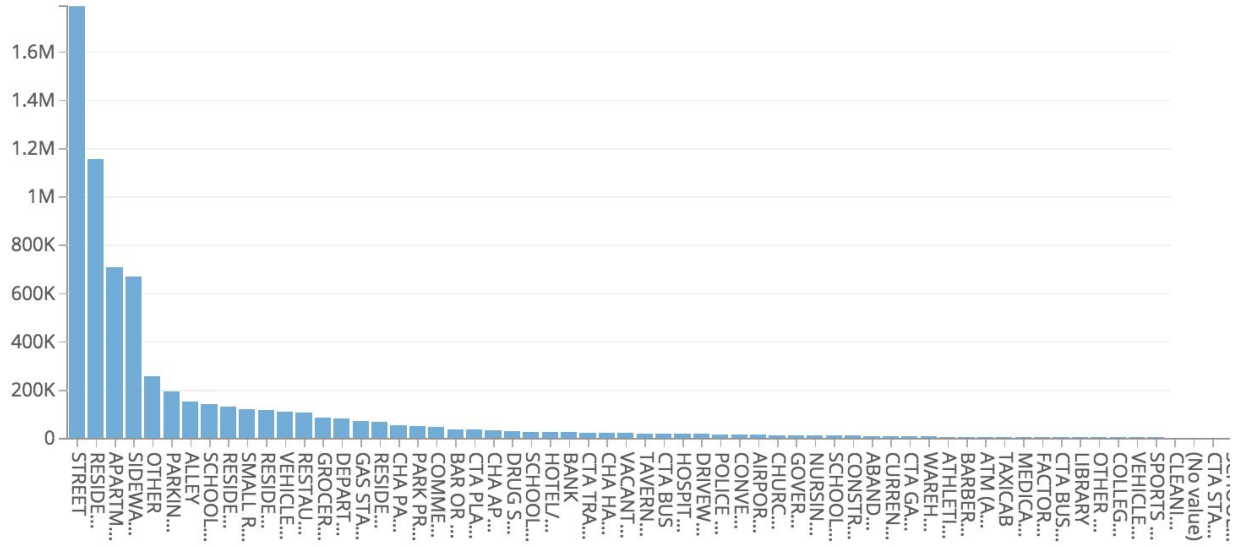
En los distritos apenas existe diferencia entre los 5 primeros, y vemos aparte del distrito 031 (centro de chicago), todos y cada uno de ellos contiene un porcentaje elevado de crímenes. Podemos indicar que el crimen no se encuentra localizado en zonas específicas si no que es un problema general de chicago. Si tuviéramos que responder con este gráfico la pregunta ¿Que zona es más segura?, la respuesta sería que si uno se pudiera permitir el centro, pues adelante aunque en caso contrario a no ser que escojas los 5 primeros, el resto están equilibrados respecto a la delincuencia.

Distrito	Cantidad de Crímenes
008	464708

011	438113
007	403275
006	394325
025	394267



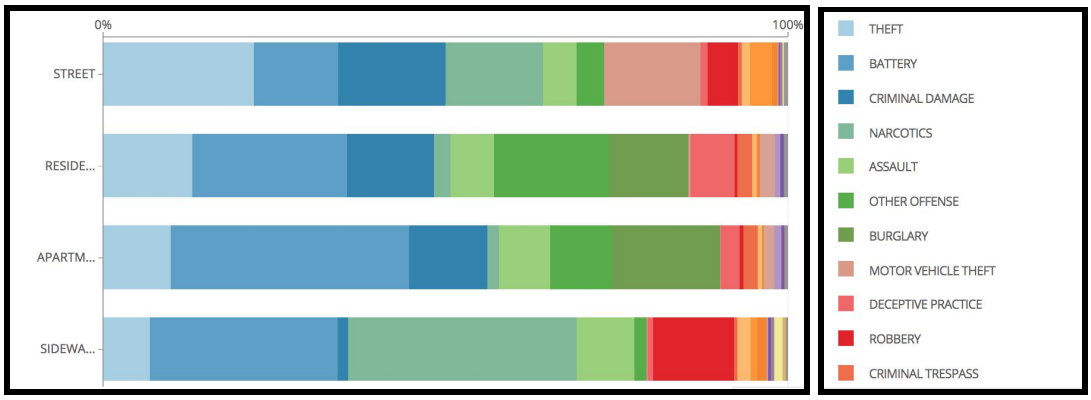
En el caso de la **location_description** se observa una mayor variación en el números de crímenes. Los 5 sitios más peligrosos son las calles (street), residencias (residence), apartamentos (apartment) y aceras (sideways). Incluso podríamos considerar calles y aceras como un mismo grupo por representar crímenes que ocurren afuera y apartamentos y residencias como otro grupo.



Localización	Crímenes
STREET	1789926
RESIDENCE	1158381
APARTMENT	710362
SIDEWALK	671339

En este caso será interesante en cómo se distribuyen los 5 tipos de crímenes más comunes dentro de los 4 lugares más comunes para ser asaltado. Para representar los resultados hemos realizado un gráfico a partir de los datos obtenidos en la consulta para así facilitar la interpretación al usuario.

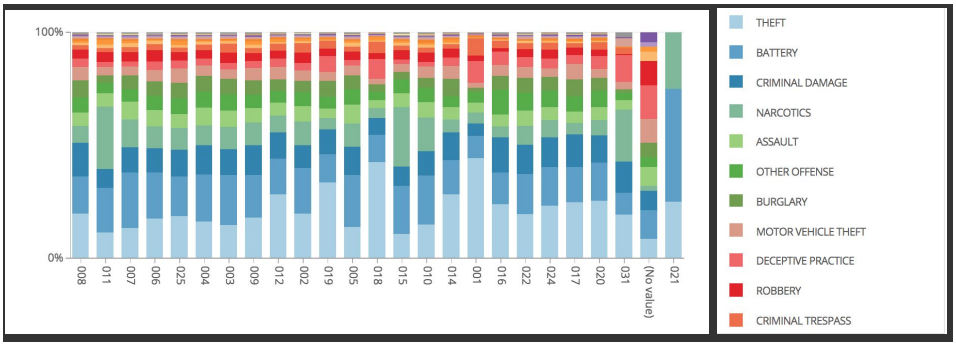
```
A = FILTER measure BY (location_description matches
'(STREET|RESIDENCE|APARTMENT|SIDEWALK)');
B = COGROUP A by (location_description, primary_type);
C = foreach B GENERATE group, COUNT(A) as count;
```



Se observa en el gráfico que en los apartamentos en donde se realizan la mayor parte de asaltos físicos (battery) . En cambio, los delitos de robo (THEFT) son más comunes en las calles.

Antes de seguir con el estudio con otros rasgos, se puede observar en el gráfico de abajo como estan distribuido los tipos de crímenes a lo largo de los distritos para comprobar si en algún distrito se suele realizar algún tipo de crimen específico.

```
A = COGROUP measure by (district, primary_type);
B = foreach A GENERATE group, primary_type, COUNT(measure) as count;
C = ORDER B BY count DESC;
dump C;
```

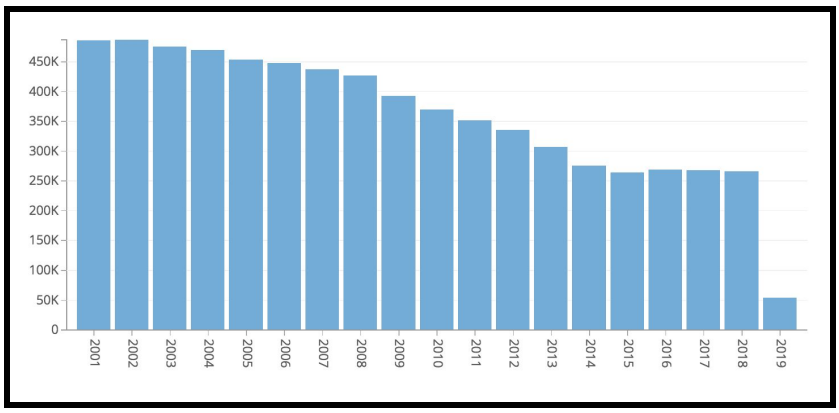


No vemos indicios que el tipo de delincuencia esté agrupado por distritos. Es cierto que en ciertos distritos (014), la criminalidad de tipo THEFT es superior al resto pero en general está distribuido de forma equilibrada a lo largo de los distritos.

A continuación vamos a profundizar en rasgos temporales para poder dar respuesta a ¿Cómo ha cambiado la delincuencia a lo largo de los años? Veamos lo que tenemos (de 2001 a 2019).

```
A = Group measure by year;
B = foreach A GENERATE group, COUNT(measure) as count;
```

Este gráfico muestra un claro patrón "periódico" de los crímenes a lo largo de muchos años. Vemos que la línea disminuye desde 2006 hasta algún punto alrededor de 2015, después de lo cual se mantiene en el mismo número de delitos hasta el 2018. Entiendo que el 2019, por ser un año reciente aún no se habrá actualizado, motivo por el cual contiene un valor tan bajo.



Año	2001	2002	2003	2004	2005	2006	2007	2008	2009
Crímenes	485754	486755	475942	469383	453716	448111	437013	427058	392683

Año	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019
Crímenes	370313	351777	335971	307105	275290	268994	268093	266232	264076	53792