

# OPI Analytics Test

Vanessa Salazar

## Section A

### Libraries

```
In [74]: %matplotlib inline
import sys

# Basic Analysis and Visualization
import pandas as pd
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
plt.style.use("ggplot")
import pandas_profiling
import seaborn as sns
import math
from datetime import timedelta

# Mapping
import folium
from folium import plugins
from folium.plugins import MarkerCluster
```

### Read input dataset

```
In [20]: data = pd.read_csv("fgj_data_opitest.csv")
data.head()
```

/usr/local/lib/python3.9/site-packages/IPython/core/interactiveshell.py:3098: DtypeWarning: Columns (15) have mixed types.Specify dtype option on import or set low\_memory=False.

```
has_raised = await self.run_ast_nodes(code_ast.body, cell_name,
```

Out[20]:

	ao_hechos	mes_hechos	fecha_hechos	ao_inicio	mes_inicio	fecha_inicio	delito	fi
0	2016.0	Enero	2016-01-31 22:16:00	2016	Febrero	2016-02-01 00:25:44	DAÑO EN PROPIEDAD AJENA INTENCIONAL	INVESTIG. EN B JL
1	2016.0	Enero	2016-01-31 20:50:00	2016	Febrero	2016-02-01 00:52:37	ROBO DE VEHICULO DE SERVICIO PARTICULAR CON VI...	INVESTIG. PA ATENCIÓ DELITO I
2	2016.0	Febrero	2016-02-01 00:30:00	2016	Febrero	2016-02-01 01:33:26	NARCOMENUDEO POSESION SIMPLE	INVESTIG. PA ATENCI NIÑOS, NI
3	2016.0	Enero	2016-01-31 22:00:00	2016	Febrero	2016-02-01 02:09:11	ROBO A TRANSEUNTE EN VIA PUBLICA CON VIOLENCIA	INVESTIG. EN IZTAP,
4	2015.0	Diciembre	2015-12-25 12:00:00	2016	Febrero	2016-02-01 02:16:49	DENUNCIA DE HECHOS	INVESTIG. EN B JL

```
In [21]: data.describe()
```

```
Out[21]:
```

	ao_hechos	ao_inicio	longitud	latitud	tempo
<b>count</b>	1.400873e+06	1.401331e+06	1.341941e+06	1.341941e+06	0.0
<b>mean</b>	2.018462e+03	2.018617e+03	-9.913714e+01	1.938701e+01	NaN
<b>std</b>	2.022233e+00	1.728106e+00	6.015875e-02	7.029166e-02	NaN
<b>min</b>	1.906000e+03	2.016000e+03	-1.002319e+02	1.909535e+01	NaN
<b>25%</b>	2.017000e+03	2.017000e+03	-9.917560e+01	1.933889e+01	NaN
<b>50%</b>	2.018000e+03	2.019000e+03	-9.914198e+01	1.938953e+01	NaN
<b>75%</b>	2.020000e+03	2.020000e+03	-9.909932e+01	1.943780e+01	NaN
<b>max</b>	2.022000e+03	2.022000e+03	-9.894686e+01	1.958333e+01	NaN

```
In [22]: df = data.copy()  
df.shape
```

```
Out[22]: (1401331, 19)
```

```
In [23]: df = df.dropna(subset=["fecha_hechos", "fecha_inicio"])  
df.shape
```

```
Out[23]: (1400870, 19)
```

```
In [24]: df["fecha_hechos"] = pd.to_datetime(df["fecha_hechos"])  
df["fecha_inicio"] = pd.to_datetime(df["fecha_inicio"])  
df = df.astype({"ao_hechos": int})
```

```
In [25]: df.drop(axis=1, columns=["tempo"], inplace=True)
```

```
In [26]: df = df[(df.ao_hechos >= 2010) & (df.ao_hechos < 2022)]
```

```
In [27]: df.shape
```

```
Out[27]: (1369570, 18)
```

## ¿Qué pruebas identificarías para asegurar la calidad de estos datos?

Al hablar de calidad en los datos es importante tomar como punto de partida los issues standard en términos de calidad que suelen presentarse y tratar de resolver aquellos que se apliquen a la características de los datos en análisis.

En este dataset en particular, observamos basicamente 3 tipos de datos:

- datetime (fecha\_hechos, fecha\_inicio)
- int64 (ao\_hechos, ao\_inicio, longitud, latitud)
- object (meses, delito y sus dependencias)

Es importante verificar la completitud del dataset, obviando los datos que contienen valores vacios, especialmente en columnas trascendentes como son fecha\_hechos, delito, alcaldia (para este ejercicio en particular).

Además, es necesario verificar la coherencia de los datos, verificar si las fechas son correctas tanto en rangos numericos permitidos, como en formato.

Verificar que los valores de latitud y longitud correspondan a la ubicacion tambien definida (alcaldia, colonia por ejemplo).

Al existir muchos campos de texto, es importante verificar la unificacion en el vocabulario, para verificar errores que pueden darse como la duplicacion de categorias por mala escritura.

Ademas, se pueden emplear herramientas disenadas para proveer reportes de analisis de los datos incluidos en el dataset, que resultan muy utiles para emitir un diagnostico, como `pandas_profiling`, por ejemplo.

## Identifica los delitos que van a la alza y a la baja en la CDMX

Para este analisis, se consideran unicamente los delitos representativos. Para esto se podrian plantear varios criterios de seleccion de delitos, sin embargo dada la premura del ejercicio, en este caso los datos representativos se eligen bajo dos consideraciones:

1. Que el delito incluya data de al menos los ultimos 5 años
2. Que el numero de ocurrencias que tienen estos delitos sea mayor a la mediana calculada sobre los datos totales.

De esta manera el numero de delitos en analisis se reduce a 110 de un total de 347.

Ademas, para determinar si un delito va a la alza o la baja, se ha calculado el pocentaje de cambio de cada delito dentro de un periodo de tiempo, para este periodo se ha tomando en consideracion los ultimos 5 anios, al ser este periodo el que incluye data mas representativa de la variacion del comportamiento de los delitos analizados. El promedio de este porcentaje de cambio en un periodo de 5 anios, definira si el delito va a la alza o a la baja.

```
In [458... def get_percentile(df_col):  
    mean_val = df_col.mean()  
    med_val = df_col.median()  
    q10 = np.percentile(df_col, 10)  
    return [round(mean_val), round(med_val), round(q10)]
```

```
In [459... relevant_crimes = []  
irrelevant_crimes = []  
for delito in df["delito"].value_counts().index:  
    years = df[df.delito == delito]["ao_hechos"].value_counts().sort_index().index  
    if (  
        (2021 in years)  
        and (2020 in years)  
        and (2019 in years)  
        and (2018 in years)  
        and (2017 in years)  
    ):  
        relevant_crimes.append(delito)  
    else:  
        irrelevant_crimes.append(delito)
```

```
In [460... df = df.loc[df["delito"].isin(relevant_crimes)]
```

```
In [461... metrics_crime = get_percentile(df.delito.value_counts())  
metrics_crime
```

Out[461]: [5839, 708, 55]

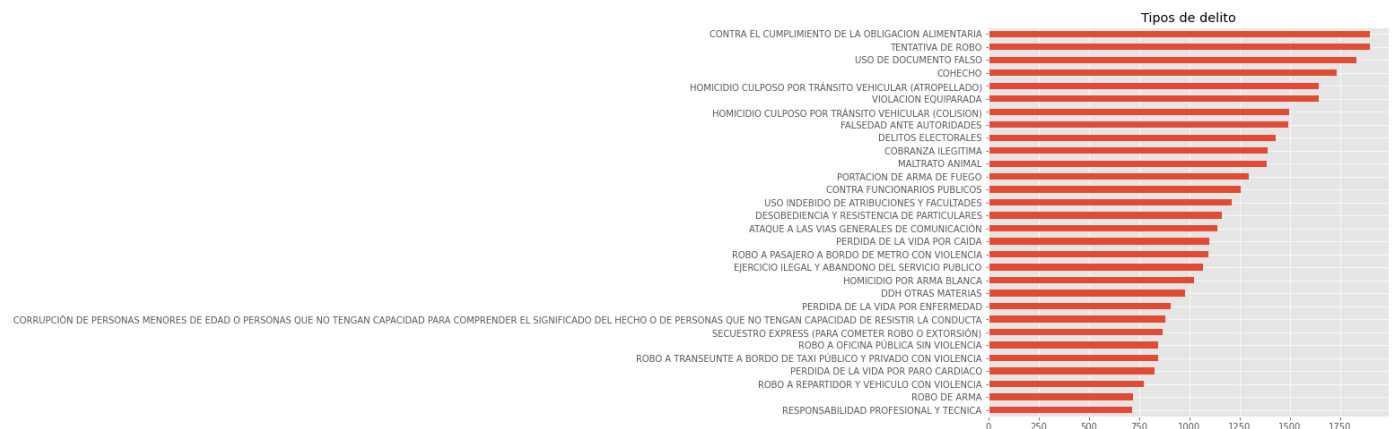
```
In [462]: len(df.groupby("delito")["delito"].count().sort_values())
```

Out[462]: 220

```
In [463]: relevant_ocurrence = (
    df.delito.value_counts().loc[lambda x: x >= metrics_crime[1]].index.tolist()
)
df = df.loc[df["delito"].isin(relevant_ocurrence)]
len(relevant_ocurrence)
```

```
In [471]: df.delito.value_counts().sort_values()[0:30].plot(
    kind="barh", title="Tipos de delito", figsize=(8, 8)
)
```

Out[471]: <AxesSubplot:title={'center': 'Tipos de delito'}>



```
In [472]: # plt.figure(figsize=(40, 40))
df_pct_change = pd.DataFrame()
for i, delito in enumerate(df.delito.value_counts().sort_values().index):
    # pos = i+1
    # plt.subplot(12, 10, pos)
    df_ = df[df.delito == delito]["ao_hechos"].value_counts().sort_index()
    pct_change = float(
        df[df.delito == delito]["ao_hechos"]
        .value_counts()
        .sort_index()
        .rename_axis("years")
        .to_frame(name="counts")[-5:]
        .pct_change()
        .counts.mean()
    )
    df_pct_change.at[delito, "pct_change"] = pct_change
    # df_.plot(title = delito[0:25])

df_pct_change_increasing = df_pct_change[df_pct_change["pct_change"] >= 0]
df_pct_change_decreasing = df_pct_change[df_pct_change["pct_change"] < 0]
```

```
In [473]: print("Lista de los crímenes en aumento, total:", df_pct_change_increasing.shape[0])
list(df_pct_change_increasing.index) # whole df
```

Lista de los crímenes en aumento, total: 73

Out[473]: ['RESPONSABILIDAD PROFESIONAL Y TECNICA',  
'PERDIDA DE LA VIDA POR PARO CARDIACO',  
'ROBO A TRANSEUNTE A BORDO DE TAXI PÚBLICO Y PRIVADO CON VIOLENCIA',  
'ROBO A OFICINA PÚBLICA SIN VIOLENCIA',  
'CORRUPCIÓN DE PERSONAS MENORES DE EDAD O PERSONAS QUE NO TENGAN CAPACIDAD PARA COMPRENDER EL SIGNIFICADO DEL HECHO O DE PERSONAS QUE NO TENGAN CAPACIDAD DE RESISTIR LA COND

UCTA',  
'PERDIDA DE LA VIDA POR ENFERMEDAD',  
'DDH OTRAS MATERIAS',  
'HOMICIDIO POR ARMA BLANCA',  
'EJERCICIO ILEGAL Y ABANDONO DEL SERVICIO PUBLICO',  
'ATAQUE A LAS VIAS GENERALES DE COMUNICACIÓN',  
'DESOBEDIENCIA Y RESISTENCIA DE PARTICULARES',  
'PORTACION DE ARMA DE FUEGO',  
'MALTRATO ANIMAL',  
'COBRANZA ILEGITIMA',  
'DELITOS ELECTORALES',  
'FALESDAD ANTE AUTORIDADES',  
'HOMICIDIO CULPOSO POR TRÁNSITO VEHICULAR (COLISION)',  
'VIOLACION EQUIPARADA',  
'COHECHO',  
'CONTRA EL CUMPLIMIENTO DE LA OBLIGACION ALIMENTARIA',  
'ROBO S/V DENTRO DE NEGOCIOS, AUTOSERVICIOS, CONVENIENCIA',  
'LESIONES CULPOSAS POR CAIDA',  
'SUSTRACCIÓN DE MENORES',  
'ROBO A PASAJERO A BORDO DE TRANSPORTE PÚBLICO SIN VIOLENCIA',  
'CONTRA LA INTIMIDAD SEXUAL',  
'ROBO A TRANSEUNTE EN PARQUES Y MERCADOS CON VIOLENCIA',  
'ROBO DE OBJETOS A ESCUELA',  
'ALLANAMIENTO DE MORADA, DESPACHO, OFICINA O ESTABLECIMIENTO MERCANTIL',  
'ROBO A TRANSEUNTE EN NEGOCIO CON VIOLENCIA',  
'PERDIDA DE LA VIDA POR SUICIDIO',  
'DAÑO EN PROPIEDAD AJENA INTENCIONAL A CASA HABITACION',  
'OMISION DE AUXILIO O DE CUIDADO',  
'ROBO A REPARTIDOR SIN VIOLENCIA',  
'ROBO A CASA HABITACION CON VIOLENCIA',  
'FALSIFICACION O ALTERACION Y USO INDEBIDO DE DOCUMENTOS',  
'ROBO A PASAJERO A BORDO DE METROBUS SIN VIOLENCIA',  
'DAÑO EN PROPIEDAD AJENA INTENCIONAL A BIENES INMUEBLES',  
'EXTORSION',  
'DISCRIMINACION',  
'LESIONES CULPOSAS',  
'PRIVACION DE LA LIBERTAD PERSONAL',  
'ACOSO SEXUAL',  
'NEGACION DEL SERVICIO PUBLICO',  
'ROBO DE PLACA DE AUTOMOVIL',  
'LESIONES CULPOSAS POR TRANSITO VEHICULAR',  
'VIOLACION',  
'ENCUBRIMIENTO',  
'NARCOMENUDEO POSESIÓN CON FINES DE VENTA, COMERCIO Y SUMINISTRO',  
'ABUSO DE AUTORIDAD Y USO ILEGAL DE LA FUERZA PUBLICA',  
'LA ADMINISTRACION DE JUSTICIA',  
'ROBO A PASAJERO A BORDO DE TRANSPORTE PÚBLICO CON VIOLENCIA',  
'DAÑO EN PROPIEDAD AJENA INTENCIONAL A AUTOMOVIL',  
'ROBO DE MOTOCICLETA SIN VIOLENCIA',  
'PERDIDA DE LA VIDA POR OTRAS CAUSAS',  
'ROBO DE VEHICULO DE PEDALES',  
'ROBO A TRANSEUNTE EN VIA PUBLICA SIN VIOLENCIA',  
'ROBO A REPARTIDOR CON VIOLENCIA',  
'PRODUCCIÓN, IMPRESIÓN, ENAJENACIÓN, DISTRIBUCIÓN, ALTERACIÓN O FALSIFICACIÓN DE TÍTULOS AL PORTADOR, DOCUMENTOS DE CRÉDITO PÚBLICOS O VALES DE CANJE',  
'ROBO A PASAJERO A BORDO DE METRO SIN VIOLENCIA',  
'TENTATIVA DE EXTORSION',  
'ROBO A PASAJERO / CONDUCTOR DE VEHICULO CON VIOLENCIA',  
'LESIONES INTENCIONALES POR GOLPES',  
'USURPACIÓN DE IDENTIDAD',  
'ABUSO SEXUAL',  
'ROBO A NEGOCIO CON VIOLENCIA',  
'NARCOMENUDEO POSESION SIMPLE',  
'DESPOJO',  
'ABUSO DE CONFIANZA',  
'ROBO DE ACCESORIOS DE AUTO',

```
'ROBO A TRANSEUNTE EN VIA PUBLICA CON VIOLENCIA',  
'AMENAZAS',  
'FRAUDE',  
'VIOLENCIA FAMILIAR']
```

```
In [474]: print("Lista de los crímenes en descenso, total:", df_pct_change_decreasing.shape[0])  
list(df_pct_change_decreasing.index)
```

Lista de los crímenes en descenso, total: 37

```
Out[474]: ['ROBO DE ARMA',  
'ROBO A REPARTIDOR Y VEHICULO CON VIOLENCIA',  
'SECUESTRO EXPRESS (PARA COMETER ROBO O EXTORSIÓN)',  
'ROBO A PASAJERO A BORDO DE METRO CON VIOLENCIA',  
'PERDIDA DE LA VIDA POR CAIDA',  
'USO INDEBIDO DE ATRIBUCIONES Y FACULTADES',  
'CONTRA FUNCIONARIOS PUBLICOS',  
'HOMICIDIO CULPOSO POR TRÁNSITO VEHICULAR (ATROPELLADO)',  
'USO DE DOCUMENTO FALSO',  
'TENTATIVA DE ROBO',  
'DAÑO EN PROPIEDAD AJENA CULPOSA POR TRÁNSITO VEHICULAR A BIENES INMUEBLES',  
'ROBO A TRANSEUNTE SALIENDO DEL BANCO CON VIOLENCIA',  
'QUEBRANTAMIENTO DE SELLOS',  
'DELITOS AMBIENTALES',  
'TORTURA',  
'ROBO DE VEHICULO DE SERVICIO PÚBLICO SIN VIOLENCIA',  
'LESIONES INTENCIONALES',  
'ROBO A PASAJERO A BORDO DE PESERO COLECTIVO CON VIOLENCIA',  
'ROBO DE VEHICULO DE SERVICIO PÚBLICO CON VIOLENCIA',  
'HOMICIDIO POR ARMA DE FUEGO',  
'ROBO DE MOTOCICLETA CON VIOLENCIA',  
'LESIONES INTENCIONALES POR ARMA BLANCA',  
'ROBO DE DOCUMENTOS',  
'ROBO DE DINERO',  
'LESIONES INTENCIONALES POR ARMA DE FUEGO',  
'DAÑO EN PROPIEDAD AJENA CULPOSA',  
'ROBO A TRANSEUNTE DE CELULAR SIN VIOLENCIA',  
'ROBO DE VEHICULO DE SERVICIO PARTICULAR CON VIOLENCIA',  
'DAÑO EN PROPIEDAD AJENA INTENCIONAL',  
'LESIONES CULPOSAS POR TRANSITO VEHICULAR EN COLISION',  
'DAÑO EN PROPIEDAD AJENA CULPOSA POR TRÁNSITO VEHICULAR A AUTOMOVIL',  
'ROBO A CASA HABITACION SIN VIOLENCIA',  
'ROBO DE VEHICULO DE SERVICIO PARTICULAR SIN VIOLENCIA',  
'ROBO DE OBJETOS DEL INTERIOR DE UN VEHICULO',  
'DENUNCIA DE HECHOS',  
'ROBO A NEGOCIO SIN VIOLENCIA',  
'ROBO DE OBJETOS']
```

**¿Cuál es la alcaldía que más delitos tiene y cuál es la que menos? ¿Por qué crees que sea esto?**

En el dataset en analisis existe un total de 514 alcaldias.

La alcaldia con el mayor numero de crímenes es:

```
In [981]: print(  
    "alcaldia: ",  
    df["alcaldia_hechos"].value_counts().idxmax(),  
    "No. delitos:",  
    df["alcaldia_hechos"].value_counts().max(),  
)
```

alcaldia: CUAUHTEMOC No. delitos: 192644

Por otro lado, existe un total de 223 alcaldias con un numero de delitos minimo, es decir 1:

```
In [989]: lower_crime_rate_alcaldias = df["alcaldia_hechos"].value_counts() == 1
lower_crime_rate_alcaldias = list(
    lower_crime_rate_alcaldias[lower_crime_rate_alcaldias == True].index
)
lower_crime_rate_alcaldias
```

```
Out[989]: ['ESPERANZA',
'TANCANHUITZ',
'MARAVATIO',
'GUELATAO DE JUAREZ',
'PUTLA VILLA DE GUERRERO',
'TECPAN DE GALEANA',
'XOCHICOATLAN',
'NANACAMILPA DE MARIANO ARISTA',
'CHAPA DE MOTA',
'XICO',
'COPALA',
'MIAHUATLAN DE PORFIRIO DIAZ',
'XICOTEPEC',
'TETEPANGO',
'TEXCALYACAC',
'SAN DIEGO DE LA UNION',
'ACATZINGO',
'PUNGARABATO',
'SAIN ALTO',
'COSAMALOAPAN DE CARPIO',
'IGNACIO DE LA LLAVE',
'TLAQUEPAQUE',
'TIQUICHEO DE NICOLAS ROMERO',
'ZAPOTITLAN PALMAS',
'TALA',
'HALACHO',
'ECUANDUREO',
'CHIGNAHUAPAN',
'TULCINGO',
'PISAFLORES',
'PETLALCINGO',
'CABO CORRIENTES',
'SANTA MARIA COLOTEPEC',
'SANTO TOMAS',
'TICUL',
'MENDEZ',
'QUIMIXTLAN',
'SAN SIMON DE GUERRERO',
'ASUNCION NOCHIXTLAN',
'CUAPIAXTLA DE MADERO',
'SAN SALVADOR EL VERDE',
'ALPATLAHUAC',
'SAN JOSE ITURBIDE',
'CHICONTEPEC',
'AHUEHUETITLA',
'TIXTLA DE GUERRERO',
'BADIRAGUATO',
'SAN JOSE CHIAPA',
'HUAUTLA DE JIMENEZ',
'LA MISION',
'NAVOJOA',
'LA MAGDALENA TLALTELULCO',
'QUECHULTENANGO',
'HUEJUTLA DE REYES',
'OZUMBA',
'ZIMATLAN DE ALVAREZ',
'SOLTEPEC',
'PINOTEPA DE DON LUIS',
'QUERENDARO',
```

'POLOTITLAN',  
'CUITZEO',  
'DELICIAS',  
'AMATLAN DE LOS REYES',  
'CONCEPCION BUENAVISTA',  
'ZIRACUARETIRO',  
'ZACAZONAPAN',  
'TEPATITLAN DE MORELOS',  
'COMAPA',  
'SANTIAGO TULANTEPEC DE LUGO GUERRERO',  
'SAN JUAN BAUTISTA ATATLAHUCA',  
'CUAUTEPEC DE HINOJOSA',  
'TEMOAYA',  
'CHAPULHUACAN',  
'GENERAL HELIODORO CASTILLO',  
'SAN FELIPE TEJALAPAM',  
'APAXCO',  
'FRANCISCO I. MADERO',  
'SANTA ANA',  
'CHURINTZIO',  
'AMOZOC',  
'OMITLAN DE JUAREZ',  
'TANHUATO',  
'COATEPEC',  
'OLUTA',  
'ARIO',  
'CIENEGA DE FLORES',  
'CHIAPA DE CORZO',  
'PEROTE',  
'AQUILES SERDAN',  
'RAFAEL DELGADO',  
'ERONGARICUARO',  
'EL SALTO',  
'ESPAÑITA',  
'TENANGO DE DORIA',  
'SAN FRANCISCO HUEHUETLAN',  
'MISANTLA',  
'TEHUITZINGO',  
'AGUA PRIETA',  
'ZITLALTEPEC DE TRINIDAD SANCHEZ SANTOS',  
'ALMOLOYA DEL RIO',  
'MINERAL DEL MONTE',  
'SAN JUAN GUICHICOVI',  
'NOGALES',  
'ACULCO',  
'HUAUCHINANGO',  
'LERDO',  
'CAÑADA MORELOS',  
'MOROLEON',  
'CALERA',  
'AMAXAC DE GUERRERO',  
'TETELA DEL VOLCAN',  
'TETELA DE OCAMPO',  
'SANTA CRUZ XOXOCOTLAN',  
'ATLAUTLA',  
'MATAMOROS',  
'SAHUAYO',  
'PAPALOTLA',  
'CHILON',  
'PARRAS',  
'EDUARDO NERI',  
'BUCTZOTZ',  
'ALVARADO',  
'CERRO DE SAN PEDRO',  
'CALPULALPAN',  
'EL FUERTE',



'SANTA ANA MAYA',  
'TOLCAYUCA',  
'CATAZAJA',  
'MACUSPANA',  
'ZINAPÉCUARO',  
'IXTAPAN DEL ORO',  
'ZACAPU',  
'TLAPEHUALA',  
'ALMOLOYA DE ALQUISIRAS',  
'ZAPOTLAN DE JUAREZ',  
'CALVILLO',  
'JALACINGO',  
'JOCOTITLAN',  
'EPAZOYUCAN',  
'TLACHICHILCO',  
'JALTIPAN',  
'TAPALPA',  
'LA PIEDAD',  
'EL ORO',  
'ATOYAC DE ALVAREZ',  
'VALLE DE SANTIAGO',  
'ARMERIA',  
'TEPETLAXOCTOC',  
'ENCARNACION DE DIAZ',  
'EBANO',  
'TEPETITLA DE LARDIZABAL',  
'SALVADOR ALVARADO',  
'RODEO',  
'MIGUEL ALEMAN',  
'ABALA',  
'PASO DEL MACHO',  
'PARAISO',  
'KANASIN',  
'HUATABAMPO',  
'HUIMANGUILLO',  
'AMANALCO',  
'TECAMACHALCO',  
'ASIENTOS',  
'MARIANO ESCOBEDO',  
'NAUPAN',  
'ATOTONILCO EL GRANDE',  
'SAN PEDRO JICAYAN',  
'TECATE',  
'HUIXTLA',  
'SAN ANDRES CHOLULA',  
'MEDELLIN',  
'ATZITZINTLA',  
'AHUMADA',  
'LIBRES',  
'SAN GREGORIO ATZOMPA',  
'PATZCUARO',  
'SAN MATIAS TLALANCALECA',  
'SENGUIO',  
'SAN JUAN DE LOS LAGOS',  
'CHARO',  
'SAN BARTOLO COYOTEPEC',  
'YURIRIA',  
'ALTAMIRA',  
'MULEGE',  
'AMECA',  
'MUZQUIZ',  
'LOS REYES DE JUAREZ',  
'MAGDALENA',  
'JESUS MARIA',  
'SABINAS',  
'JALOSTOTITLAN',

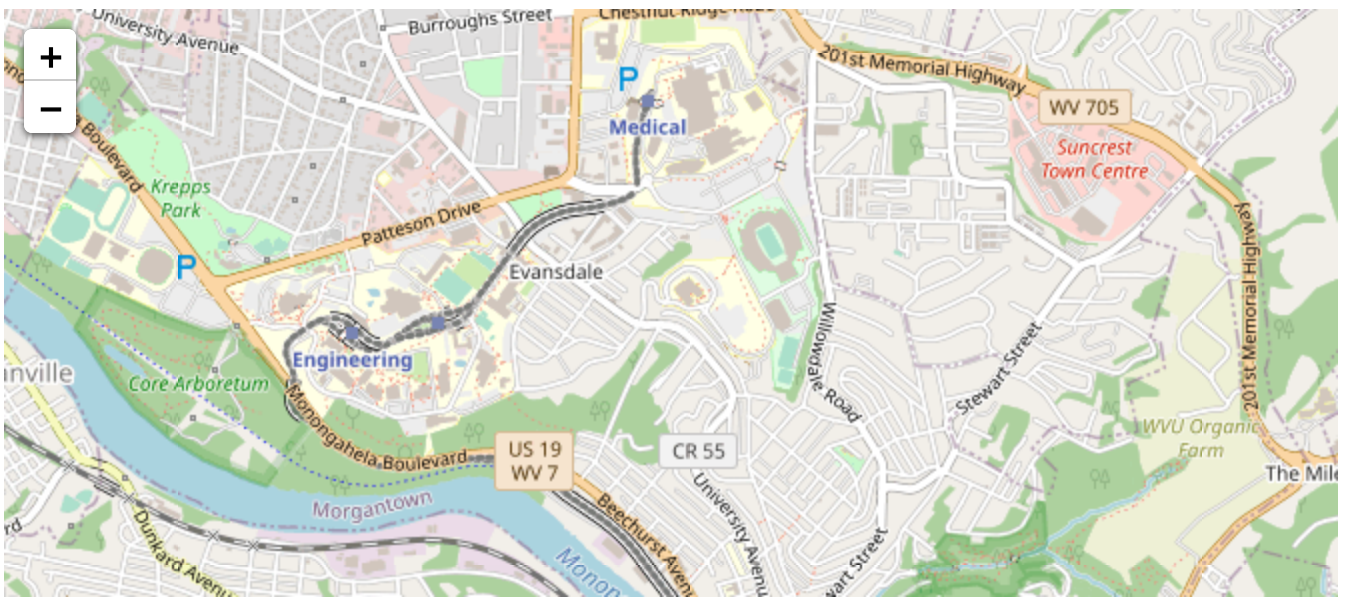
'SAN PABLO DEL MONTE',  
'HUEYPOXTLA',  
'TLAXCO',  
'COYUCA DE CATALAN',  
'TEOTITLAN DE FLORES MAGON',  
'CADEREYTA JIMENEZ',  
'COYUCA DE BENITEZ',  
'ZAPOTLAN EL GRANDE',  
'SAN FRANCISCO DEL RINCON',  
'XONACATLAN',  
'SAN MIGUEL XOXTLA',  
'TANCITARO',  
'MATIAS ROMERO',  
'PROGRESO DE OBREGON',  
'VALLE HERMOSO',  
'TEPEACA',  
'CARRILLO PUERTO',  
'TUXTLA GUTIERREZ',  
'JARAL DEL PROGRESO',  
'SAN PEDRO POCHUTLA',  
'CUITLAHUAC',  
'CHAPULTEPEC',  
'TOTOLAPAN',  
'AHUATLAN',  
'ALTO LUCERO DE GUTIERREZ BARRIOS',  
'BATOPILAS',  
'SAN PEDRO ATOYAC',  
'ACONCHI',  
'ATITALAQUIA',  
'COMPOSTELA',  
'HUEYAPAN DE OCAMPO',  
'CEDRAL']

```
In [328... df_cuauhtemoc = df[df["alcaldia_hechos"] == "CUAUHTEMOC"]
```

```
In [329... df_cauhtemoc = df_cauhtemoc.dropna()
m = folium.Map([39.645, -79.96], zoom_start=14)
for index, row in df_cauhtemoc.iterrows():
    folium.CircleMarker(
        [row["latitud"], row["longitud"]],
        radius=3,
        popup=row["delito"],
        fill_color="#3db7e4",
    ).add_to(m)
```

In [331... m

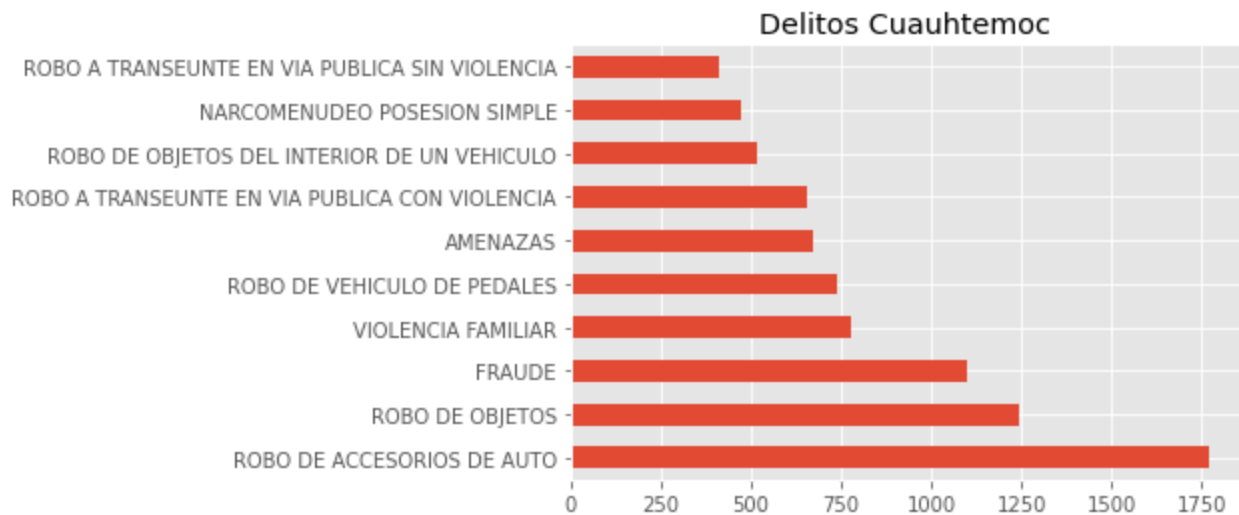
Out[331]:





```
In [103... df_cuauhtemoc["delito"].value_counts().sort_values(ascending=False)[0:10].plot(
    kind="barh", title="Delitos Cuauhtemoc"
)
```

Out[1032]: <AxesSubplot:title={'center': 'Delitos Cuauhtemoc'}>



```
In [103... df_cuauhtemoc["colonia_hechos"].value_counts().head()
```

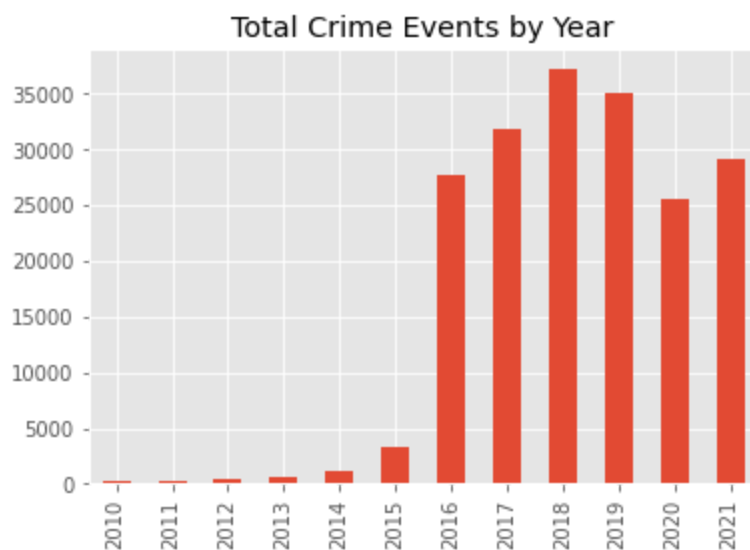
Out[1035]:

CENTRO	2676
ROMA NORTE	1412
DOCTORES	958
GUERRERO	791
MORELOS	780

Name: colonia\_hechos, dtype: int64

```
In [993... df_cuauhtemoc["ao_hechos"].value_counts().sort_index().plot(
    kind="bar", title="Total Crime Events by Year"
)
```

Out[993]: <AxesSubplot:title={'center': 'Total Crime Events by Year'}>



He realizado un breve de los delitos cometidos en la alcaldia de Cuauhtemoc, habiendo encontrado lo

siguiente:

- El crimen en la alcaldia empezo a aumentar significativamente en el año 2016.
- Los delitos con mayor ocurrencia estan vinculados al robo (vehiculos, objetos, accesorios de vehiculos asi como tambien transeuntes)
- Otro crímenes tambien destacados son la narcoposesion y el maltrato intrafamiliar.
- La latitud y longitud del dataset muestra una zona con presencia de muchos comercios, la zona parece bastante turistica al estar formada por colonias que contienen sitios de interes para visitantes, pero tambien se observan centros nocturnos, prostibulos, etc.

Con esta informacion puedo suponer que Cuauhtemoc corresponde a una alcaldia de nivel socioeconomico bajo donde hay mas probabilidad delincuencia debido a la presencia de comercios en el sector y de visitantes interesados en la zona, ademas del narcomenudeo que puede ser tambien causante de que delitos que causa que se considere a esta como la alcaldia de mayor delincuencia en el periodo que se analiza.

Tras una breve investigacion, he confirmado que Cuauhtemoc es efectivamente una zona de nivel socioeconomico popular/bajo, en donde se encuentran las colonias consideradas mas peligrosas en la CDMX como son Centro, Sta. Maria la Ribera, Morelos, etc. Este sector tiene problematicas como la seguridad pública, recolección de basura, hacinamiento, vivienda, desempleo, insalubridad, drogadiccion, desercion escolar etc. Estos factores indudablemente inciden en la tasa de delitos alta que se puede evidenciar, ademas de justificar los delitos de violencia registrados debido a las condiciones de vida de las personas de esta alcaldia.

Por otro lado, las alcaldias con menor numero de crímenes corresponden en su mayoria a alcaldias alejadas con un numero reducido de habitantes y menor extension, lo que explica la tasa baja de delitos en estas zonas.

fuelle : <https://www.redalyc.org/pdf/325/32515208.pdf>

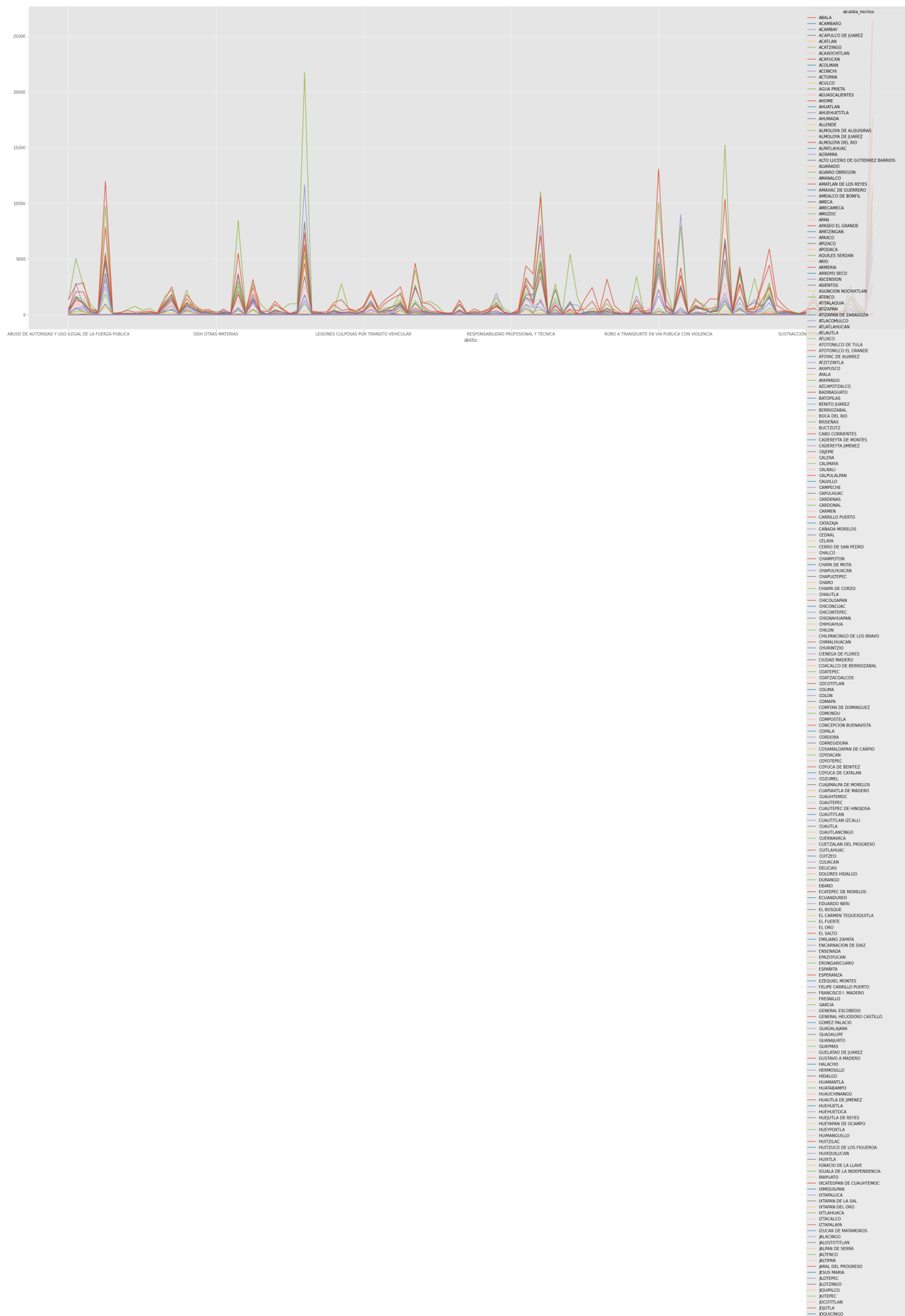
**¿Cuáles son los delitos que más caracterizan a cada alcaldía? Es decir, delitos que suceden con mayor frecuencia en una alcaldía y con menor frecuencia en las demás.**

El analisis realizado para esta seccion, incluye graficas de cada delito Vs. el numero de ocurrencias del mismo para cada alcaldia. Analiticamente se ha determinado tambien el delito con mas frecuencia para cada alcaldia, siendo predominantes las alcaldias de 'Cuauhtemoc' e 'Iztapalapa', para la mayoria de delitos analizados. Este resultado es consistente con el analizado previamente, el mismo que categorizaba a Cuauhtemoc como la alcaldia con mayor indice de delincuencia en el periodo analizado.

```
In [28]: df_crime_frequency = (  
    df.groupby("delito")["alcaldia_hechos"].value_counts().reset_index(name="counts")  
)  
# for i, crime in enumerate(df_crime_frequency.delito.unique()):  
#     pos = i + 1  
#     d = df_crime_frequency[df_crime_frequency.delito == crime]  
#     d.plot(  
#         x="alcaldia_hechos", y="counts", figsize=(10, 4), grid=True, title=crime[0:30]  
#     )  
#     plt.xticks(rotation=90)
```

```
In [325... table_crime_frecuency = pd.crosstab(df["delito"], df["alcaldia_hechos"]).T
```

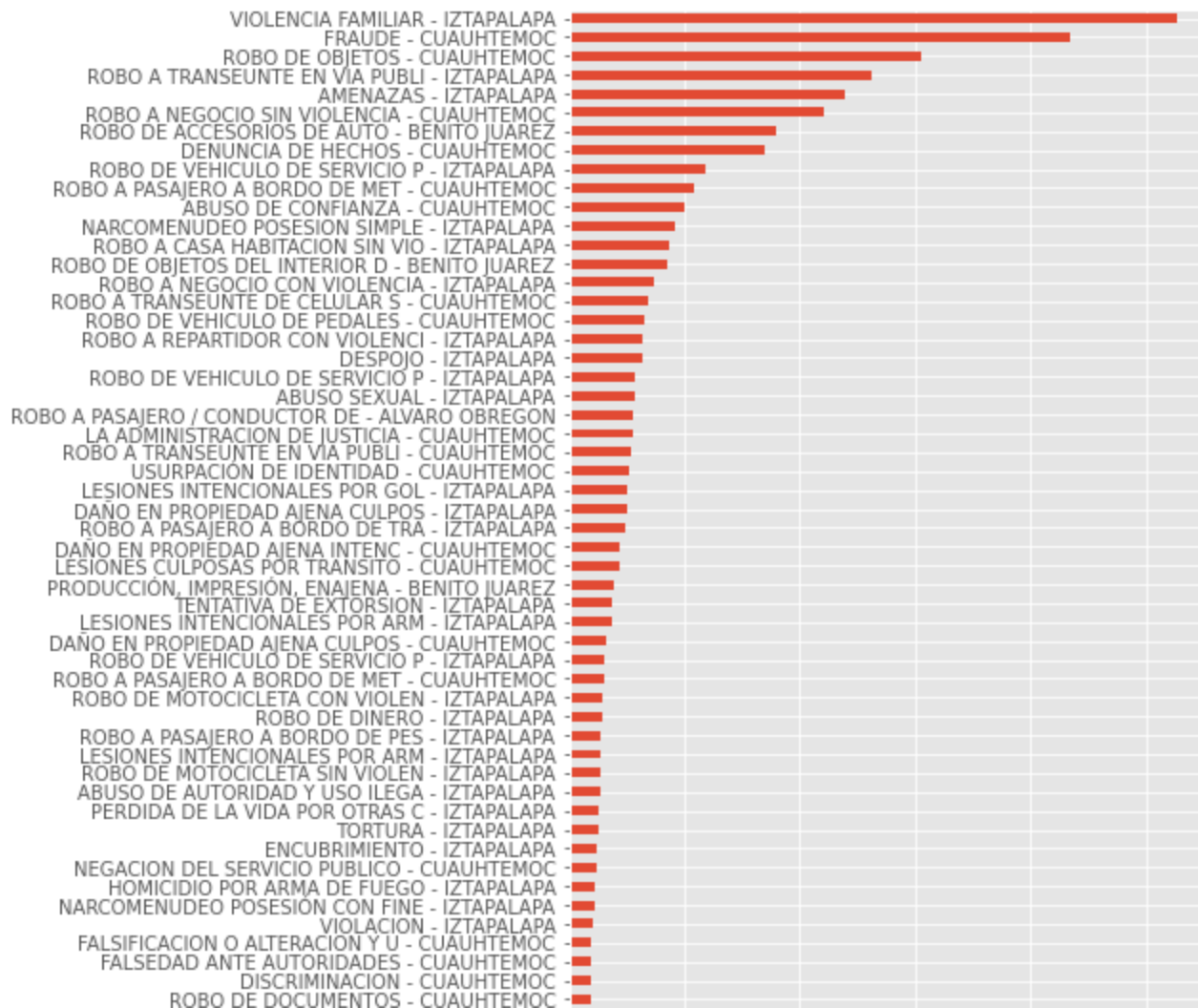
```
Out[325]: <AxesSubplot:xlabel='delito'>
```



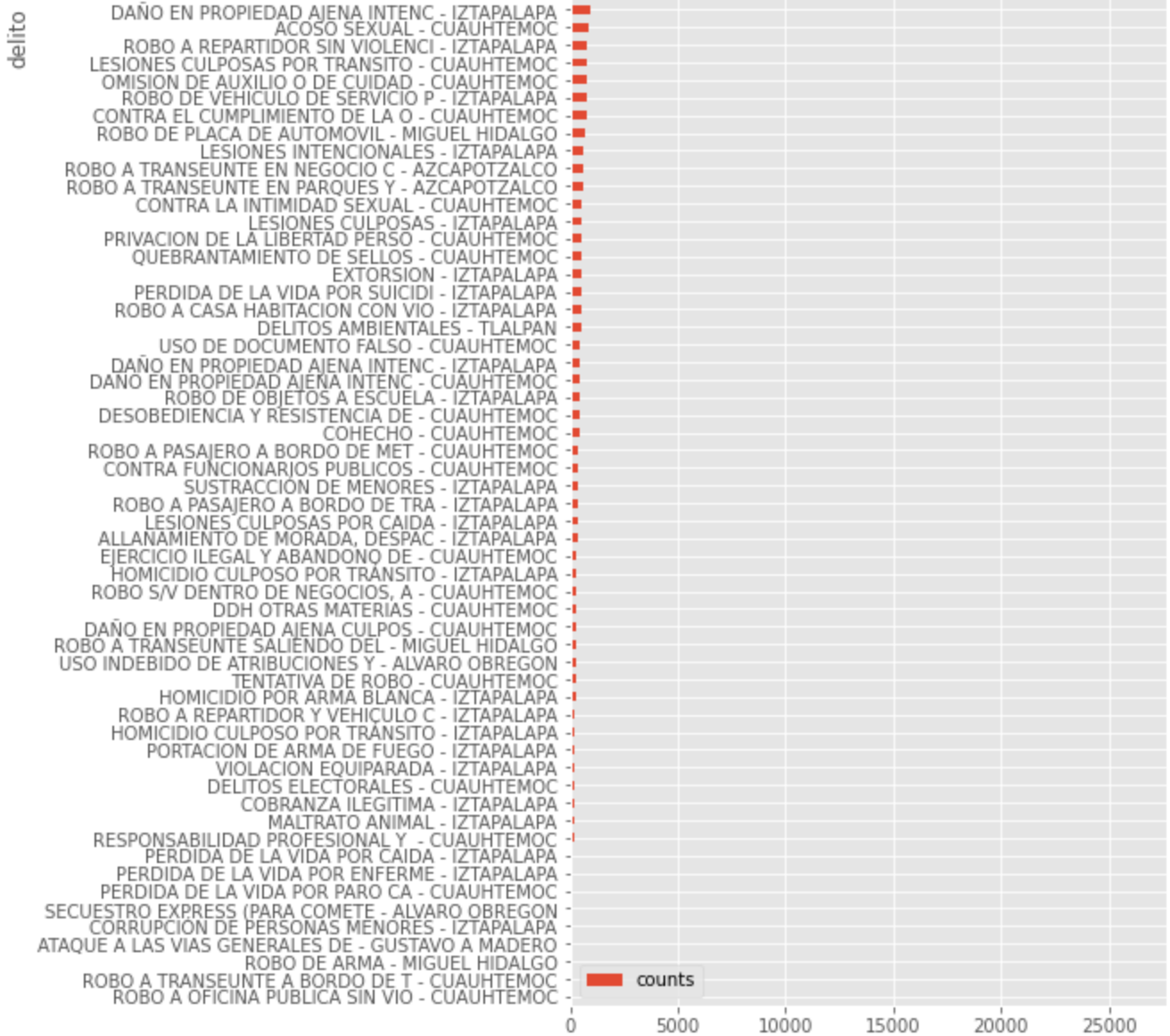
[illegible]

```
In [342... df_alcaldia_rate = pd.DataFrame(crime_rate_max, columns=["delito", "counts"])
df_alcaldia_rate.sort_values(by=["counts"]).plot(
    x="delito", y="counts", kind="barh", figsize=(6, 20)
)
```

```
Out[342]: <AxesSubplot:ylabel='delito'>
```







Diseña un indicador que mida el nivel de “inseguridad”. Genéralo al nivel de desagregación que te parezca más adecuado (ej. manzana, calle, AGEB, etc.). Analiza los resultados ¿Encontraste algún patrón interesante? ¿Qué decisiones se podrían tomar con el indicador?

```
In [47]: def show_distribution(var_data):
    """
    This function will make a distribution (graph) and display it
    """

    # Get statistics
    min_val = var_data.min()
    max_val = var_data.max()
    mean_val = var_data.mean()
    med_val = var_data.median()
    mod_val = var_data.mode()[0]
    q25_val = np.percentile(var_data, 25)
    q99_val = np.percentile(var_data, 99)

    print(
        "Minimum:{:.2f}\nMean:{:.2f}\nMedian:{:.2f}\nMode:{:.2f}\nMaximum:{:.2f}\nnp25:{:.2f}\nnp99:{:.2f}"
        .format(min_val, mean_val, med_val, mod_val, max_val, q25_val, q99_val)
    )

    # Create a figure for 2 subplots (2 rows, 1 column)
    fig, ax = plt.subplots(2, 1, figsize=(15, 5))
```



```

# Plot the histogram
ax[0].hist(var_data, color="navy")
ax[0].set_ylabel("Frequency")

# Add lines for the mean, median, and mode
ax[0].axvline(x=q25_val, color="yellow", linestyle="dashed", linewidth=2)
ax[0].axvline(x=q99_val, color="yellow", linestyle="dashed", linewidth=2)
ax[0].axvline(x=mean_val, color="cyan", linestyle="dashed", linewidth=2)
ax[0].axvline(x=med_val, color="red", linestyle="dashed", linewidth=2)

# Plot the boxplot
ax[1].boxplot(var_data, vert=False)
ax[1].set_xlabel("Engagement")

# Add a title to the Figure
fig.suptitle("Data Distribution")

# Show the figure
fig.show()
return mean_val, q99_val

def get_normalized_data(df):
    q25 = np.percentile(df.counts, 25)
    q75 = np.percentile(df.counts, 75)
    cut_off = (q75 - q25) * 3 # iqr * k
    lower = q25 - cut_off
    upper = q75 + cut_off
    df_ = df[df.counts > 1]
    df_norm = df_[df_.counts < upper]
    return df_norm

```

El indicador a ser usado pretende medir el nivel de inseguridad en la CDMX que considera para su calculo el porcentaje de cambio delincucional en cada zona. Para esto, se ha planteado un analisis de percentiles, la desagregacion se ha realizado a nivel de alcaldia. Se han planteado una categorizacion delincucional en 4 niveles:

- A: nivel bajo de delincuencia
- B: nivel medio de delincuencia
- C: nivel medio-alto de delincuencia
- D: nivel alto de delincuencia

Tras eliminar outliers se obtiene la distribucion de la muestra, y se seleccionan los percentiles 50 y 95 para definir los limites de cada rango de frecuencia de delitos. El percentil 50 definira el limite superior para el nivel bajo, el percentil 95 definira el limite superior para el nivel medio, los valores a partir del percentil 95 se considerara como nivel medio-alto de delincuencia hasta el limite superior, que vendra dado tras analizar los valores de outliers encontrados en el dataset, los valores a partir de este limite se consideraran como nivel alto de delincuencia.

Ademas, para este analisis se toma en consideracion el porcentaje de cambio, es decir el porcentaje de crecimiento o decrecimiento delincucional en un periodo de tiempo de 5 anios. En caso de que el indice de delincuencia este en decrecimiento, se multiplicara la ocurrencia del delito por un valor  $k_1 < 1$ . Por otro lado, cuando el indice de delincuencia este en crecimiento se multiplicara la ocurrencia del delito por un valor  $k_2 > 1$  que viene dado por la suma de  $1 +$  indice de crecimiento.

Este ultimo threshold, se ha fijado en un valor de 1500, dado que tras analizar los datos existe un evidente aumento en la ocurrencia de delitos a partir de un valor cercano a 1500. Estos son outliers extremos que deberían ser categorizados como nivel alto de delincuencia.

El indicador pretende mostrar el nivel de inseguridad de CDMX a través del porcentaje que representan las alcaldías con nivel delictuoso alto y medio alto, del total de alcaldías analizadas.

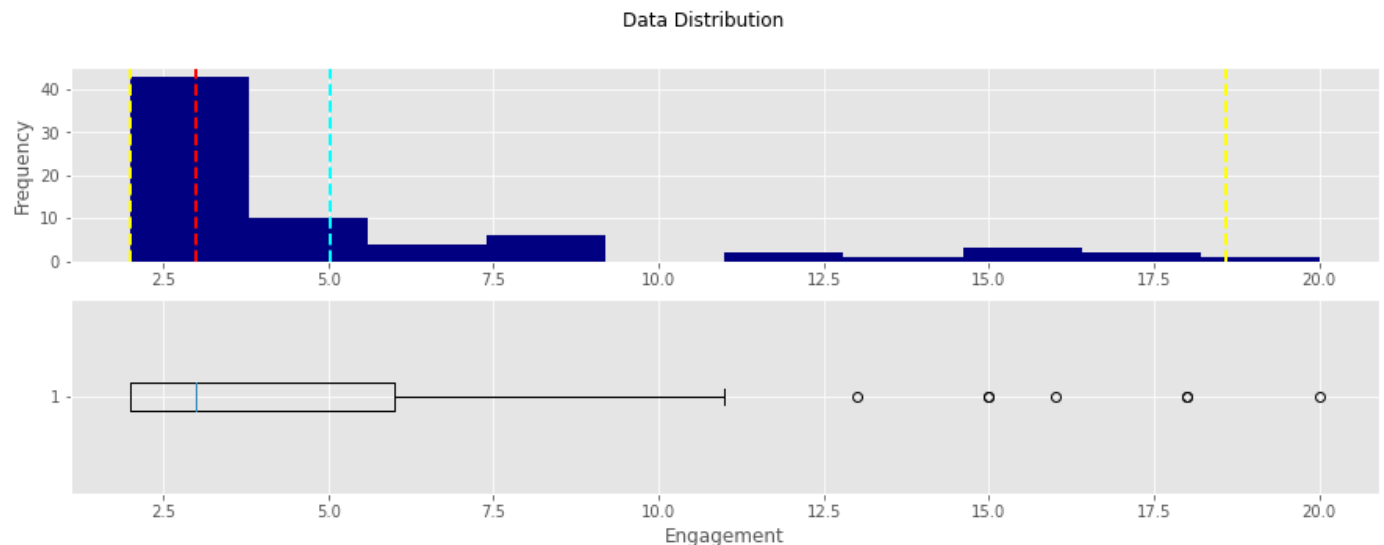
```
In [117...] df_2021 = df[df['ao_hechos']==2021]
```

```
In [118...] df_indicador = (  
    df_2021.alcaldia_hechos.value_counts().rename_axis("alcaldia").to_frame(name="counts"  
)  
df_indicador_norm = get_normalized_data(df_indicador)
```

```
In [119...] mean, q95 = show_distribution(df_indicador_norm.counts)
```

```
Minimum:2.00  
Mean:5.04  
Median:3.00  
Mode:2.00  
Maximum:20.00  
p25:2.00  
p95:18.58
```

```
/var/folders/bl/kh03pyqslknc_yg2fvfr6rvr0000gp/T/ipykernel_9837/132521588.py:42: UserWarning: Matplotlib is currently using module://matplotlib_inline.backend_inline, which is a non-GUI backend, so cannot show the figure.  
fig.show()
```



```
In [120...] df_pct_change_alcaldia = pd.DataFrame()  
for i, alcaldia in enumerate(df.alcaldia_hechos.value_counts().index):  
    pct_change_alcaldia = float(  
        df[df.alcaldia_hechos == alcaldia]["ao_hechos"]  
        .value_counts()  
        .sort_index()  
        .rename_axis("years")  
        .to_frame(name="counts")[-5:]  
        .pct_change()  
        .counts.mean()  
    )  
    df_pct_change_alcaldia.at[alcaldia, "pct_change"] = pct_change_alcaldia
```

```
In [121...] df_indicador_ = pd.concat([df_indicador, df_pct_change_alcaldia], axis=1)  
df_indicador_["pct_change"] = df_indicador_["pct_change"].fillna(0)
```

```
df_indicador_["counts"] = df_indicador_["counts"].fillna(0)
df_indicador_["trend"] = ""
for item, count in enumerate(df_indicador_["pct_change"].values):
    if count >= 0:
        df_indicador_["trend"][item] = 1 + count
    else:
        df_indicador_["trend"][item] = 1 * 0.9
```

/var/folders/bl/kh03pyqslknc\_yg2fvfr6rvr0000gp/T/ipykernel\_9837/626511551.py:9: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
df_indicador_["trend"][item] = 1 * 0.9
/var/folders/bl/kh03pyqslknc_yg2fvfr6rvr0000gp/T/ipykernel_9837/626511551.py:7: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame
```

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
df_indicador_["trend"][item] = 1 + count
```

In [122... df\_indicador\_["crime\_rate"] = df\_indicador\_.counts \* df\_indicador\_.trend

In [123... df\_indicador\_.head()

Out[123]:

	counts	pct_change	trend	crime_rate
<b>IZTAPALAPA</b>	32295.0	-0.001958	0.9	29065.5
<b>CUAUHTEMOC</b>	31102.0	-0.029687	0.9	27991.8
<b>GUSTAVO A MADERO</b>	22898.0	0.005627	1.005627	23026.835744
<b>ALVARO OBREGON</b>	15425.0	0.025970	1.02597	15825.57964
<b>BENITO JUAREZ</b>	15227.0	-0.057624	0.9	13704.3

In [124... df\_indicador\_["indicator"] = ""

```
for i, ind in enumerate(df_indicador_["crime_rate"].values):
    if ind <= mean:
        df_indicador_["indicator"][i] = "A"
    elif (ind > mean) & (ind <= q95):
        df_indicador_["indicator"][i] = "B"
    elif (ind > q95) & (ind <= 1500):
        df_indicador_["indicator"][i] = "C"
    elif ind > 1500:
        df_indicador_["indicator"][i] = "D"
```

/var/folders/bl/kh03pyqslknc\_yg2fvfr6rvr0000gp/T/ipykernel\_9837/4063198131.py:10: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
df_indicador_["indicator"][i] = "D"
/var/folders/bl/kh03pyqslknc_yg2fvfr6rvr0000gp/T/ipykernel_9837/4063198131.py:8: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame
```

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
df_indicador_["indicator"][i] = "C"
/var/folders/bl/kh03pyqslknc_yg2fvfr6rvr0000gp/T/ipykernel_9837/4063198131.py:6: Setting
```

WithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
df_indicador_["indicator"][i] = "B"
```

/var/folders/bl/kh03pyqslknc\_yg2fvfr6rvr0000gp/T/ipykernel\_9837/4063198131.py:4: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
df_indicador_["indicator"][i] = "A"
```

Tras el análisis realizado, que puede replicarse para colonias, calles, etc., se ha definido la siguiente distribución:

```
In [125... print(
    "Nivel bajo de delincuencia:",
    df_indicador_.indicator.value_counts()["A"],
    "alcaldias",
)
print(
    "Nivel medio de delincuencia:",
    df_indicador_.indicator.value_counts()["B"],
    "alcaldias",
)
print(
    "Nivel medio-alto de delincuencia:",
    df_indicador_.indicator.value_counts()["C"],
    "alcaldias",
)
print(
    "Nivel alto de delincuencia:",
    df_indicador_.indicator.value_counts()["D"],
    "alcaldias",
)
```

```
Nivel bajo de delincuencia: 498 alcaldias
Nivel medio de delincuencia: 22 alcaldias
Nivel medio-alto de delincuencia: 26 alcaldias
Nivel alto de delincuencia: 16 alcaldias
```

Como se menciona anteriormente el indicador pretende mostrar el nivel de inseguridad de CDMX, y se define por la suma del % que representan las alcaldías con nivel delincucional medio-alto + la suma del % que representan las alcaldías con nivel delincucional alto, del total de alcaldías analizadas.

En este caso, la suma del porcentaje delincucional de las alcaldías con nivel medio-alto (26) y alto (16) representan el 7.5% de total del alcaldías para el año 2021. En el año 2020 se categorizaron con nivel delincucional medio-alto a 38 alcaldías, y con nivel delincucional alto a 16 alcaldías, lo que representa el 9.67%. Este indicador nos permite demostrar que con respecto al año 2020, en el año 2021 ha habido una reducción de alcaldías con niveles de delincuencia medio-altos y altos, de esta manera y de forma general el nivel de delincuencia en CDMX ha bajado un 2.17% del año 2020 al 2021.

```
In [106... # report = pandas_profiling.ProfileReport(df)
# report
```