Exam - Fri 12, Nov 2021

November 12, 2021

Scientific Programming - Data Science Master @ University of Trento

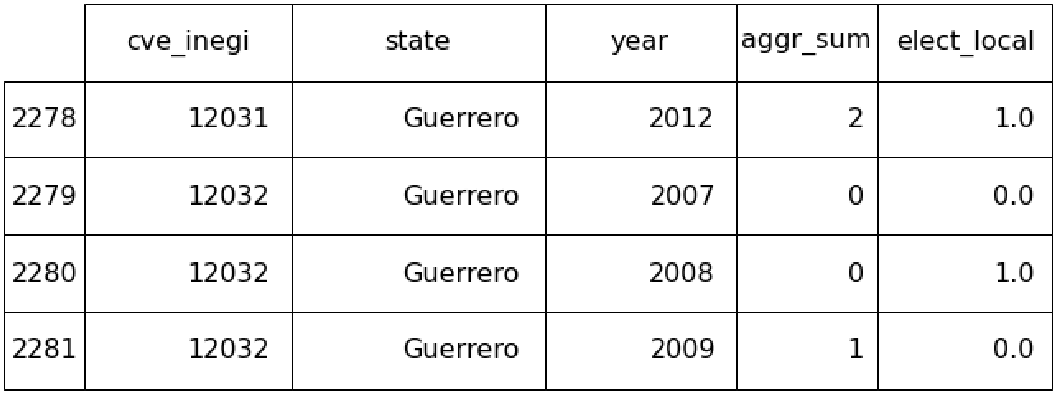
0.1 Part A - Mexican Drug Wars

Open Jupyter and start editing this notebook `exam-2021-11-12.ipynb`

0.2 Attacks during elections

In the file `Dataset_HighProfileCriminalViolence.tab` are listed the number of attacks occurred to elected officials in Mexico from years 2007 to 2012. Focus only on columns `cve_inegi`, `state`, `year`, `aggr_sum`, `elect_local`:

[2]:



	cve_inegi	state	year	aggr_sum	elect_local
2278	12031	Guerrero	2012	2	1.0
2279	12032	Guerrero	2007	0	0.0
2280	12032	Guerrero	2008	0	1.0
2281	12032	Guerrero	2009	1	0.0

- Municipalities where the attack occurred are identified by a 5 digits `cve_inegi` code: first two digits indicate the state, 3 last ones the town. **NOTE**: first file entries only have 4 digits as the leading zero is implied, take care of this case
- `aggr_sum`: number of attacks occurred in a particular municipality / year.
- `elect_local`: `1.0` if a **local** election occurred in the year of the attack (ignore other `elect_*`)

0.3 load_mexico

Extract Mexican state codes, names, the counts of attacks, and the years when **local** elections occurred, and RETURN a dictionary of dictionaries mapping **two digit** state codes **as strings** to the extracted info.

- use `csv.DictReader` with `delimiter='\t'` and `utf8` encoding (municipalities will look weird but we don't use them)
- use exactly 6 cells for `attacks` lists: assume all were carried out between `2007`and `2012` included
- **DO NOT** assume the years in rows repeat with a pattern, for example municipality `21132` has two successive 2012 years!

[3]:

```python
import csv

def load(filename):
    raise Exception('TODO IMPLEMENT ME !')

mexico_db = load('Dataset_HighProfileCriminalViolence.tab')
mexico_db
```

Complete expected output can be found in `expected_mexico_db.py`

[4]:

```
```

```
EXERPT (note keys order doesn't matter):

{
  '08': {
              'attacks': [0, 5, 7, 12, 7, 2],
              'local_election_years': [2007, 2010],
              'state_code': '08',
              'state_name': 'Chihuahua'
             }
  '12': {
              'attacks': [4, 11, 11, 9, 3, 10],
              'local_election_years': [2008, 2011, 2012],
              'state_code': '12',
              'state_name': 'Guerrero'
             }
  .
  .
  .
}
```

[5]:

```python
# TESTING
from pprint import pformat; from expected_mexico_db import expected_mexico_db
for sid in expected_mexico_db.keys():
    if sid not in mexico_db: print('\nERROR: MISSING state', sid); break
    for k in expected_mexico_db[sid]:
        if k not in mexico_db[sid]:
            print('\nERROR at state', sid,'\n\n   MISSING key:', k); break
        if expected_mexico_db[sid][k] != mexico_db[sid][k]:
            print('\nERROR at state', sid, 'key:',k)
            print('  ACTUAL:\n', pformat(mexico_db[sid][k]))
            print('  EXPECTED:\n', pformat(expected_mexico_db[sid][k]))
            break
if len(mexico_db) > len(expected_mexico_db):
    print('ERROR! There are more states than expected!')
    print('  ACTUAL:\n', len(mexico_db))
    print('  EXPECTED:\n', len(expected_mexico_db))
```

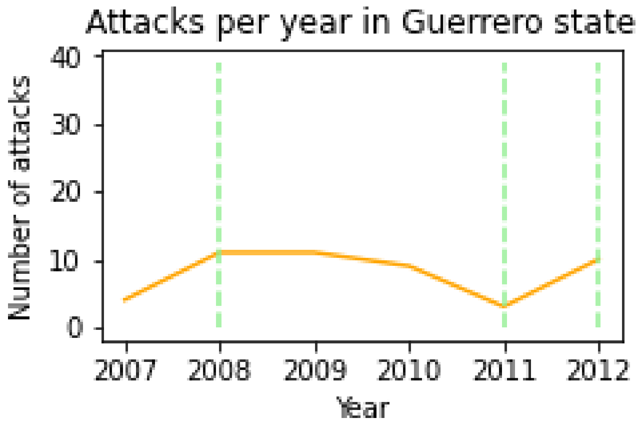
0.4 show_attacks

Given a `state_code` and , display a chart of the attacks count over the years.

- normalize the height so to have all charts as high as the maximum possible attack count in the db
- show vertical dashed lines in proximity of election years (use `linestyle='dashed'`), using the same color
- you are allowed to use constants for years

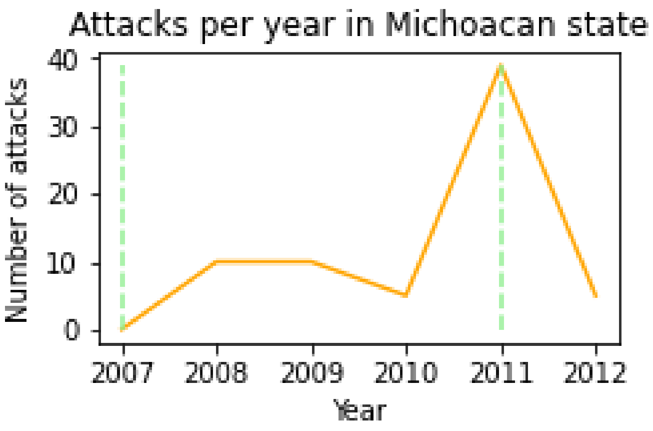
[6]:
```
%matplotlib inline
import matplotlib.pyplot as plt

def show_attacks(state_code, mexdb):
    raise Exception('TODO IMPLEMENT ME !')

show_attacks('12', mexico_db)   # Guerrero
```

```
max attacks happened in any state: 39
```



```
[7]: show_attacks('16', mexico_db)    # Michoacan
```

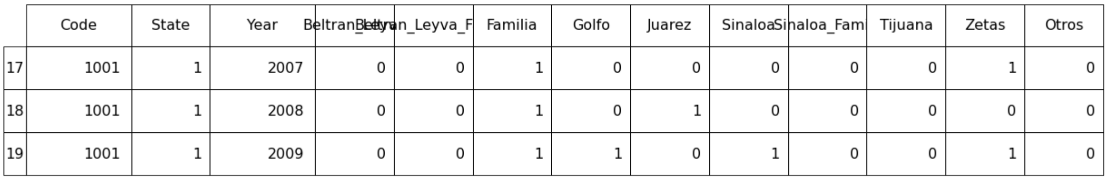
```
max attacks happened in any state: 39
```



0.5 Cartels

In the file CosciaRios2012_DataBase.csv are listed attacks performed by criminal organizations (cartels) in various years. For each row, the columns from 3-12 have a 1 if the corresponding cartel named in the header was involved in the attack, and 0 otherwise. Example:

```
[8]:
```



	Code	State	Year	Beltran_Leyva	Beltran_Leyva_F	Familia	Golfo	Juarez	Sinaloa	Sinaloa_Fam	Tijuana	Zetas	Otros
17	1001	1	2007	0	0	1	0	0	0	0	0	1	0
18	1001	1	2008	0	0	1	0	1	0	0	0	0	0
19	1001	1	2009	0	0	1	1	0	1	0	0	1	0

Write a function which given a `filename` and a `year`, processes the dataset and RETURN a dictionary mapping cartel names to a list of **sorted** states (no duplicates) where the cartel performed attacks in the given `year`.

- use a `csv.reader` with `utf8` encoding
- pick state code from `State` column and state names from previous `mexico_db` (you only need names) - if missing put state code (i.e. `09`)
- **NOTE**: `Sinaloa` is a special case, since it is both a state and a cartel.

[9]:

```
import csv
def cartels(filename, mexdb, year):
    raise Exception('TODO IMPLEMENT ME !')

cartels2003 = cartels('CosciaRios2012_DataBase.csv', mexico_db, 2003)
assert cartels2003['Juarez'] == ['Durango', 'Guerrero', 'Jalisco', 'Sinaloa',␣
 ↪'Tamaulipas']
assert cartels2003['Familia'] == []
assert cartels2003['Sinaloa'] == ['Jalisco', 'Mexico', 'Nayarit', 'Nuevo Leon',␣
 ↪'Sinaloa', 'Sonora', 'Tamaulipas']
from pprint import pprint
pprint(cartels2003, width=120)
```

```
{'Beltran_Leyva': ['Morelos', 'Sinaloa'],
 'Beltran_Leyva_Family': [],
 'Familia': [],
 'Golfo': ['Durango', 'Mexico', 'Nuevo Leon', 'San Luis Potosi', 'Tamaulipas',
'Veracruz', 'Yucatan'],
 'Juarez': ['Durango', 'Guerrero', 'Jalisco', 'Sinaloa', 'Tamaulipas'],
 'Otros': [],
 'Sinaloa': ['Jalisco', 'Mexico', 'Nayarit', 'Nuevo Leon', 'Sinaloa', 'Sonora',
'Tamaulipas'],
 'Sinaloa_Family': ['Guerrero'],
 'Tijuana': [],
 'Zetas': []}
```

[11]:

```
#cartels('CosciaRios2012_DataBase.csv', mexico_db, 2004)
```

[ ]:

```
```