Taller 7

Métodos Computacionales para Políticas Públicas - URosario

Entrega: viernes 7-abr-2017 11:59 PM

[David Valles]

[david.valles@urosario.edu.co]

Instrucciones:

- Guarde una copia de este *Jupyter Notebook* en su computador, idealmente en una carpeta destinada al material del curso.
- Modifique el nombre del archivo del *notebook*, agregando al final un guión inferior y su nombre y apellido, separados estos últimos por otro guión inferior. Por ejemplo, mi *notebook* se llamaría: mcpp_taller7_santiago_matallana
- Marque el *notebook* con su nombre y e-mail en el bloque verde arriba. Reemplace el texto "[Su nombre acá]" con su nombre y apellido. Similar para su e-mail.
- Desarrolle la totalidad del taller sobre este *notebook*, insertando las celdas que sea necesario debajo de cada pregunta. Haga buen uso de las celdas para código y de las celdas tipo *markdown* según el caso.
- Recuerde salvar periódicamente sus avances.
- Cuando termine el taller:
  1. Descárguelo en PDF. Si tiene algún problema con la conversión, descárguelo en HTML.
  2. Suba todos los archivos a su repositorio en GitHub, en una carpeta destinada exclusivamente para este taller, antes de la fecha y hora límites.

(Todos los ejercicios tienen el mismo valor.)

Este taller tiene dos partes. Una obligatoria, relativamente fácil, y otra voluntaria y más retadora. Los invito a intentar desarrollar el taller en su totalidad.

En este taller exploraremos los datos de crimen de Chicago.

Descargue los datos de crimen del Chicago Data Portal solo para el año 2015 (https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present/ijzp-q8t2 (https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present/ijzp-q8t2)).

```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        %matplotlib inline
        plt.rcParams["figure.figsize"] = [18.0, 10.0]
        plt.style.use('ggplot')
```

```
In [158]: crimes = pd.read_csv('Crimes_-_2001_to_present.csv')
```

In [85]:

```
crimes
```

```
Out[85]:
```

	ID	Case Number	Date	Block	IUCR	Primary Type	Description
0	10514462	HZ256372	01/01/2015 12:00:00 AM	073XX S EXCHANGE AVE	0281	CRIM SEXUAL ASSAULT	NON-AGGRAVATED
1	10515175	HZ257172	11/24/2015 05:30:00 PM	033XX W ADAMS ST	0820	THEFT	$500 AND UNDER
2	10077106	HY266148	05/19/2015 01:12:00 AM	009XX W BELMONT AVE	0560	ASSAULT	SIMPLE
3	10301916	HY469211	01/01/2015 12:00:00 AM	062XX W BARRY AVE	0266	CRIM SEXUAL ASSAULT	PREDATORY
4	10160687	HY313819	06/24/2015 06:00:00 AM	052XX N MARMORA AVE	1811	NARCOTICS	POSS: CANNABIS 30GMS OR LESS
5	10181647	HY215976	04/09/2015 04:30:51 PM	061XX S CHAMPLAIN AVE	2024	NARCOTICS	POSS: HEROIN(WHITE)
6	10192026	HY360910	07/29/2015 04:21:21 PM	029XX W FLOURNOY ST	2027	NARCOTICS	POSS: CRACK
7	10203133	HY381916	08/14/2015 10:55:00 AM	054XX W WALTON ST	1821	NARCOTICS	MANU/DEL:CANNABIS 10GM OR LESS
8	10211492	HY257237	05/12/2015 11:54:19 AM	066XX S WASHTENAW AVE	1811	NARCOTICS	POSS: CANNABIS 30GMS OR LESS
9	10211589	HY340025	07/14/2015 06:00:00 AM	033XX W FILLMORE ST	1812	NARCOTICS	POSS: CANNABIS MORE THAN 30GMS
10	10211597	HY334821	07/10/2015 06:00:00 AM	006XX W OHARE ST	2093	NARCOTICS	FOUND SUSPECT NARCOTICS
11	10211611	HY333584	07/09/2015 06:00:00 AM	033XX W FILLMORE ST	1812	NARCOTICS	POSS: CANNABIS MORE THAN 30GMS
12	10211617	HY333517	07/09/2015 06:00:00 AM	033XX W FILLMORE ST	1812	NARCOTICS	POSS: CANNABIS MORE THAN 30GMS
13	10211621	HY369125	08/04/2015 05:24:00 PM	075XX S STONY ISLAND AVE	2017	NARCOTICS	MANU/DELIVER:CRACK
14	10211623	HY331028	07/07/2015 06:00:00 AM	033XX W FILLMORE ST	1812	NARCOTICS	POSS: CANNABIS MORE THAN 30GMS

Parte obligatoria

1.

Calcule el número de crímenes en cada Community Area en 2015. Haga un gráfico de barras que lo ilustre.

```
In [98]: crimes_by_community = crimes.groupby('Community Area')
         crimes_by_community
```

```
Out[98]: <pandas.core.groupby.DataFrameGroupBy object at 0x000000C1A28F0208>
```

In [99]:

```
crimes_by_community.groups
```

```
Out[99]: {0: [15933, 195914],
          1: [146,
           159,
           168,
           433,
           550,
           676,
           703,
           716,
           757,
           837,
           880,
           1057,
           1114,
           1146,
           1159,
           1199,
           1229,
           1395,
           1399,
           1461,
           1655,
           1703,
           1880,
           1967,
           2323,
           2348,
           2489,
           2490,
           2536,
           2622,
           2648,
           2669,
           2685,
           2686,
           2755,
           2771,
           2774,
           2882,
           2898,
           2927,
           3147,
           3150,
           3190,
           3266,
           3408,
           3492,
           3547,
           3844,
           3851,
           3889,
           3951,
           4052,
           4073,
           4082,
           4282,
           4362,
           4534,
           4543,
           4625,
           4846,
           4889,
           4902,
           4915,
```

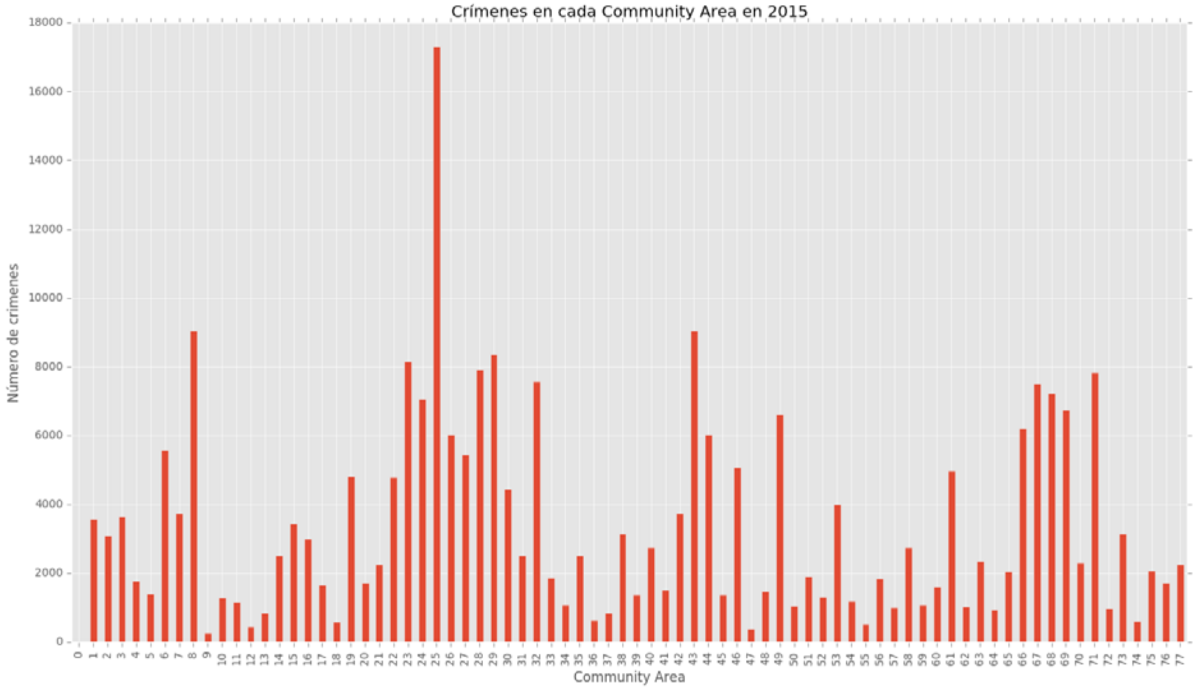
In [100]:

```
community_crime_count = crimes_by_community['ID'].agg('count')
community_crime_count.to_frame()
```

Out[100]:

	ID
Community Area	
0	2
1	3579
2	3100
3	3652
4	1767
5	1392
6	5578
7	3751
8	9050
9	257
10	1273
11	1155
12	446
13	841
14	2523
15	3437
16	3010
17	1654
18	586
19	4821
20	1727
21	2257
22	4790
23	8167
24	7062
25	17311
26	6018
27	5436
28	7920
29	8367
...	...
48	1478
49	6614
50	1045
51	1920
52	1316

```
In [101]: plt.title("Crímenes en cada Community Area en 2015")
          plt.ylabel("Número de crimenes")
          community_crime_count.plot(kind='bar');
```

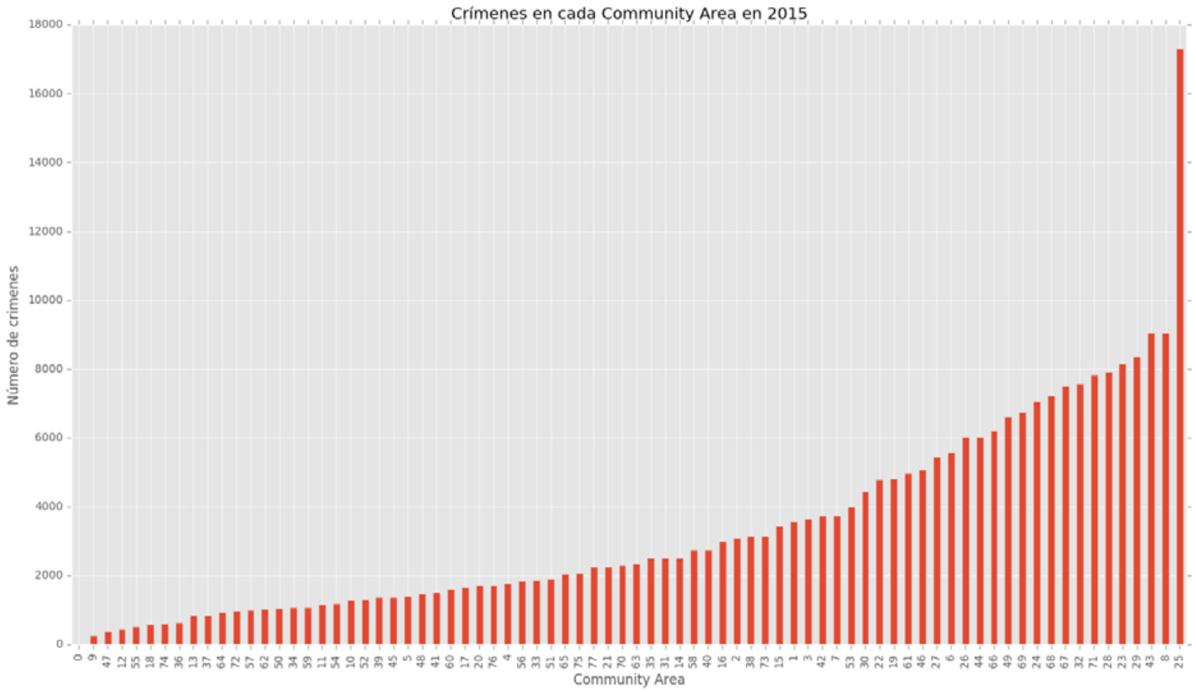


2.

Ordene las Community Areas de acuerdo con el número de crímenes. ¿Qué Community Area (por nombre, idealmente) presenta el mayor número de crímenes? ¿El menor?

```
In [102]: community_crime_count.sort()
          plt.title("Crímenes en cada Community Area en 2015")
          plt.ylabel("Número de crimenes")
          community_crime_count.plot(kind='bar');
```

```
C:\Users\PC\Anaconda3\lib\site-packages\ipykernel\__main__.py:1: FutureWarning:
sort is deprecated, use sort_values(inplace=True) for INPLACE sorting
  if __name__ == '__main__':
```



El Community Area que presentó mas crimenes fue el 25 (Austin) con 17311 crimenes, mientras que el que presentó menos crimenes fue el 9 (Edison Park) con 257. Existen dos datos que corresponden al Community Area "0", pero este no existe.

3.

Cree una tabla cuyas filas sean días del año (yyyy-mm-dd) y las columnas las 77 Community Areas. En cada campo de la tabla deberá haber el correspondiente número de crímenes. Seleccione algunas Community Areas que le llamen la atención y haga un gráfico de serie de tiempo.

Pista: El siguiente código puede serle útil.

# Create function to strip time from date field, and use it to create another column def to_day(timestamp): return timestamp.replace(minute=0,hour=0, second=0) crimes['Day'] = crimes['Date'].apply(to_day)

In [133]:

```
crimes['Created Date'] = pd.to_datetime(crimes['Date']).apply(lambda x: x.date())
crimes['Created Date']
```

```
Out[133]: 0         2015-01-01
          1         2015-11-24
          2         2015-05-19
          3         2015-01-01
          4         2015-06-24
          5         2015-04-09
          6         2015-07-29
          7         2015-08-14
          8         2015-05-12
          9         2015-07-14
          10        2015-07-10
          11        2015-07-09
          12        2015-07-09
          13        2015-08-04
          14        2015-07-07
          15        2015-07-01
          16        2015-08-04
          17        2015-08-04
          18        2015-08-06
          19        2015-08-06
          20        2015-08-06
          21        2015-08-05
          22        2015-08-05
          23        2015-08-05
          24        2015-08-07
          25        2015-08-13
          26        2015-08-14
          27        2015-08-13
          28        2015-04-24
          29        2015-04-15
                       ...
          263131    2015-04-01
          263132    2015-07-01
          263133    2015-02-08
          263134    2015-09-22
          263135    2015-09-24
          263136    2015-03-30
          263137    2015-09-19
          263138    2015-12-27
          263139    2015-08-01
          263140    2015-12-03
          263141    2015-07-09
          263142    2015-12-17
          263143    2015-11-15
          263144    2015-10-17
          263145    2015-10-30
          263146    2015-01-06
          263147    2015-01-28
          263148    2015-02-07
          263149    2015-03-09
          263150    2015-02-01
          263151    2015-07-23
          263152    2015-06-11
          263153    2015-06-26
          263154    2015-10-17
          263155    2015-08-14
          263156    2015-11-01
          263157    2015-03-01
          263158    2015-08-02
          263159    2015-02-01
          263160    2015-02-01
          Name: Created Date, dtype: object
```

In [147]:

```
crimes_by_community_created_date = crimes.groupby(['Community Area','Created Date'
])
```

In [149]:

```
crimes_by_community_day_count = crimes_by_community_created_date['ID'].agg('count'
)
```

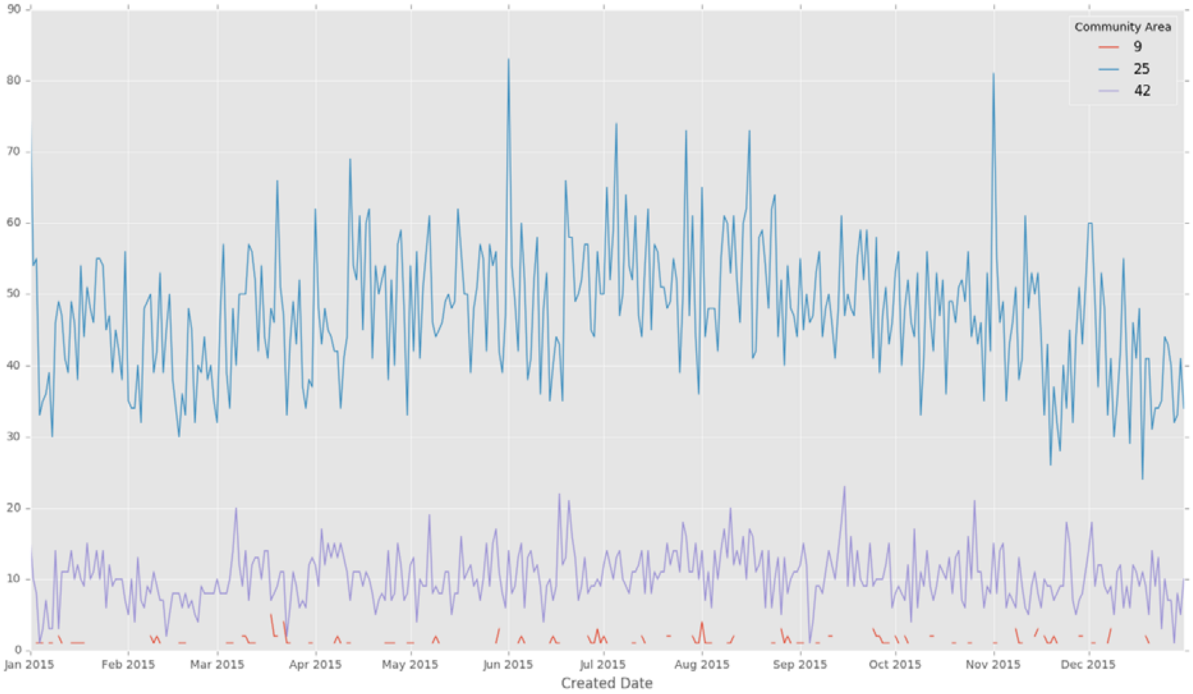
In [151]:

```
community_arrest_timeseries = crimes_by_community_day_count.unstack('Community Area')
community_arrest_timeseries
```

Out[151]:

Community Area	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**...**	**68**	**69**	**70**	**71**	**72**	**73**
Created Date																	
2015-01-01	NaN	15.0	9.0	13.0	6.0	5.0	23.0	12.0	47.0	1.0	...	29.0	25.0	9.0	45.0	2.0	8.0
2015-01-02	NaN	5.0	9.0	8.0	3.0	2.0	10.0	9.0	27.0	NaN	...	12.0	22.0	5.0	17.0	1.0	11.0
2015-01-03	NaN	7.0	11.0	9.0	7.0	4.0	6.0	11.0	27.0	1.0	...	23.0	12.0	8.0	18.0	NaN	8.0
2015-01-04	NaN	12.0	7.0	9.0	10.0	3.0	15.0	5.0	16.0	1.0	...	13.0	15.0	9.0	12.0	1.0	5.0
2015-01-05	NaN	6.0	7.0	5.0	4.0	5.0	15.0	7.0	11.0	1.0	...	16.0	12.0	8.0	17.0	NaN	5.0
2015-01-06	NaN	8.0	8.0	6.0	5.0	NaN	14.0	7.0	13.0	NaN	...	15.0	14.0	6.0	11.0	2.0	8.0
2015-01-07	NaN	6.0	2.0	5.0	6.0	1.0	8.0	6.0	17.0	1.0	...	11.0	7.0	4.0	16.0	4.0	7.0
2015-01-08	NaN	6.0	7.0	3.0	5.0	NaN	6.0	5.0	8.0	1.0	...	9.0	9.0	6.0	10.0	2.0	4.0
2015-01-09	NaN	10.0	5.0	10.0	2.0	4.0	14.0	6.0	21.0	NaN	...	18.0	14.0	10.0	20.0	1.0	9.0
2015-01-10	NaN	6.0	12.0	8.0	NaN	1.0	10.0	5.0	24.0	2.0	...	9.0	13.0	6.0	28.0	3.0	3.0
2015-01-11	NaN	8.0	6.0	11.0	5.0	4.0	20.0	4.0	26.0	1.0	...	17.0	8.0	11.0	17.0	2.0	10.0
2015-01-12	NaN	3.0	6.0	6.0	6.0	1.0	7.0	11.0	17.0	NaN	...	12.0	18.0	6.0	19.0	3.0	5.0
2015-01-13	NaN	10.0	10.0	9.0	5.0	4.0	8.0	6.0	15.0	NaN	...	19.0	12.0	9.0	11.0	2.0	6.0
2015-01-14	1.0	17.0	8.0	9.0	4.0	3.0	6.0	11.0	15.0	1.0	...	21.0	16.0	6.0	24.0	NaN	5.0
2015-01-15	NaN	9.0	8.0	8.0	6.0	7.0	10.0	11.0	18.0	1.0	...	20.0	20.0	4.0	22.0	3.0	7.0
2015-01-16	NaN	13.0	6.0	12.0	5.0	5.0	12.0	12.0	22.0	1.0	...	18.0	17.0	8.0	16.0	4.0	5.0
2015-01-17	NaN	12.0	5.0	5.0	2.0	5.0	16.0	7.0	31.0	1.0	...	29.0	13.0	7.0	15.0	2.0	6.0
2015-01-18	NaN	12.0	6.0	12.0	7.0	5.0	14.0	8.0	18.0	1.0	...	20.0	22.0	3.0	20.0	1.0	11.0
2015-01-19	NaN	3.0	12.0	7.0	3.0	5.0	10.0	11.0	25.0	NaN	...	24.0	19.0	3.0	28.0	NaN	13.0
2015-01-20	NaN	8.0	8.0	9.0	10.0	3.0	13.0	12.0	31.0	NaN	...	19.0	15.0	5.0	23.0	1.0	10.0
2015-01-21	NaN	18.0	8.0	9.0	10.0	9.0	12.0	9.0	29.0	2.0	...	11.0	12.0	7.0	29.0	5.0	3.0
2015-01-22	NaN	7.0	7.0	6.0	9.0	7.0	14.0	10.0	22.0	NaN	...	19.0	13.0	11.0	22.0	1.0	5.0
2015-01-23	NaN	12.0	11.0	6.0	4.0	6.0	8.0	6.0	27.0	1.0	...	27.0	18.0	6.0	20.0	1.0	7.0
2015-01-24	NaN	14.0	9.0	7.0	8.0	4.0	11.0	15.0	23.0	NaN	...	10.0	13.0	5.0	17.0	1.0	8.0
2015-01-25	NaN	5.0	9.0	3.0	5.0	2.0	20.0	6.0	29.0	NaN	...	15.0	13.0	7.0	15.0	3.0	4.0
2015-01-26	NaN	10.0	8.0	5.0	4.0	3.0	13.0	5.0	13.0	1.0	...	18.0	17.0	10.0	29.0	1.0	8.0
2015-01-27	NaN	7.0	8.0	6.0	3.0	3.0	6.0	3.0	13.0	NaN	...	20.0	16.0	9.0	20.0	2.0	11.0
2015-01-28	NaN	15.0	9.0	11.0	1.0	1.0	9.0	11.0	26.0	NaN	...	13.0	16.0	3.0	18.0	2.0	7.0
2015-01-29	NaN	9.0	9.0	11.0	2.0	3.0	11.0	7.0	26.0	1.0	...	19.0	20.0	4.0	24.0	2.0	6.0
2015-01-30	NaN	8.0	11.0	6.0	5.0	3.0	12.0	10.0	22.0	NaN	...	20.0	22.0	6.0	20.0	5.0	7.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2015-12-02	NaN	12.0	6.0	11.0	4.0	7.0	12.0	11.0	23.0	1.0	...	23.0	13.0	6.0	24.0	1.0	8.0
2015-12-03	NaN	6.0	11.0	10.0	2.0	5.0	12.0	8.0	32.0	1.0	...	17.0	19.0	7.0	17.0	2.0	7.0
2015-12-04	NaN	8.0	9.0	7.0	5.0	5.0	10.0	4.0	27.0	NaN	...	19.0	15.0	3.0	22.0	2.0	7.0

```
In [153]: community_arrest_timeseries[[9,25,42]].plot();
```



Parte voluntaria

Descargue la base de datos de información socioeconómica (https://data.cityofchicago.org/Health-Human-Services/Census-Data-Selected-socioeconomic-indicators-in-C/kn9c-c2s2 (https://data.cityofchicago.org/Health-Human-Services/Census-Data-Selected-socioeconomic-indicators-in-C/kn9c-c2s2)).

```
In [216]: crimes1 = pd.read_csv('Census_Data_-_Selected_socioeconomic_indicators_in_Chicago_
          _2008___2012.csv')
          crimes1
```

Out[216]:

	Community Area Number	**COMMUNITY AREA NAME**	**PERCENT OF HOUSING CROWDED**	**PERCENT HOUSEHOLDS BELOW POVERTY**	**PERCENT AGED 16+ UNEMPLOYED**	**PERCENT AGED 25+ WITHOUT HIGH SCHOOL DIPLOMA**	**PERCENT AGED UNDER 18 OR OVER 64**	**PER**
0	1.0	Rogers Park	7.7	23.6	8.7	18.2	27.5	239
1	2.0	West Ridge	7.8	17.2	8.8	20.8	38.5	230
2	3.0	Uptown	3.8	24.0	8.9	11.8	22.2	357
3	4.0	Lincoln Square	3.4	10.9	8.2	13.4	25.5	375
4	5.0	North Center	0.3	7.5	5.2	4.5	26.2	571
5	6.0	Lake View	1.1	11.4	4.7	2.6	17.0	600
6	7.0	Lincoln Park	0.8	12.3	5.1	3.6	21.5	715
7	8.0	Near North Side	1.9	12.9	7.0	2.5	22.6	886
8	9.0	Edison Park	1.1	3.3	6.5	7.4	35.3	409
9	10.0	Norwood Park	2.0	5.4	9.0	11.5	39.5	328
10	11.0	Jefferson Park	2.7	8.6	12.4	13.4	35.5	277
11	12.0	Forest Glen	1.1	7.5	6.8	4.9	40.5	441
12	13.0	North Park	3.9	13.2	9.9	14.4	39.0	265
13	14.0	Albany Park	11.3	19.2	10.0	32.9	32.0	213
14	15.0	Portage Park	4.1	11.6	12.6	19.3	34.0	243
15	16.0	Irving Park	6.3	13.1	10.0	22.4	31.6	272
16	17.0	Dunning	5.2	10.6	10.0	16.2	33.6	262
17	18.0	Montclaire	8.1	15.3	13.8	23.5	38.6	220
18	19.0	Belmont Cragin	10.8	18.7	14.6	37.3	37.3	154
19	20.0	Hermosa	6.9	20.5	13.1	41.6	36.4	150
20	21.0	Avondale	6.0	15.3	9.2	24.7	31.0	200
21	22.0	Logan Square	3.2	16.8	8.2	14.8	26.2	319
22	23.0	Humboldt park	14.8	33.9	17.3	35.4	38.0	137
23	24.0	West Town	2.3	14.7	6.6	12.9	21.7	431
24	25.0	Austin	6.3	28.6	22.6	24.4	37.9	159
25	26.0	West Garfield Park	9.4	41.7	25.8	24.5	43.6	109
26	27.0	East Garfield	8.2	42.4	19.6	21.3	43.2	129

4.

Cree una tabla que agregue el número de crímenes por Community Area. Una esa tabla con la de datos socioeconómicos y cree un "scatter plot" de número de crímenes vs ingreso per cápita. Explique la relación en palabras.

```
In [203]: crimes1
```

Out[203]:

	Community Area Number	COMMUNITY AREA NAME	PERCENT OF HOUSING CROWDED	PERCENT HOUSEHOLDS BELOW POVERTY	PERCENT AGED 16+ UNEMPLOYED	PERCENT AGED 25+ WITHOUT HIGH SCHOOL DIPLOMA	PERCENT AGED UNDER 18 OR OVER 64	PER
0	1.0	Rogers Park	7.7	23.6	8.7	18.2	27.5	239
1	2.0	West Ridge	7.8	17.2	8.8	20.8	38.5	230
2	3.0	Uptown	3.8	24.0	8.9	11.8	22.2	357
3	4.0	Lincoln Square	3.4	10.9	8.2	13.4	25.5	375
4	5.0	North Center	0.3	7.5	5.2	4.5	26.2	571
5	6.0	Lake View	1.1	11.4	4.7	2.6	17.0	600
6	7.0	Lincoln Park	0.8	12.3	5.1	3.6	21.5	715
7	8.0	Near North Side	1.9	12.9	7.0	2.5	22.6	886
8	9.0	Edison Park	1.1	3.3	6.5	7.4	35.3	409
9	10.0	Norwood Park	2.0	5.4	9.0	11.5	39.5	328
10	11.0	Jefferson Park	2.7	8.6	12.4	13.4	35.5	277
11	12.0	Forest Glen	1.1	7.5	6.8	4.9	40.5	441
12	13.0	North Park	3.9	13.2	9.9	14.4	39.0	265
13	14.0	Albany Park	11.3	19.2	10.0	32.9	32.0	213
14	15.0	Portage Park	4.1	11.6	12.6	19.3	34.0	243
15	16.0	Irving Park	6.3	13.1	10.0	22.4	31.6	272
16	17.0	Dunning	5.2	10.6	10.0	16.2	33.6	262
17	18.0	Montclaire	8.1	15.3	13.8	23.5	38.6	220
18	19.0	Belmont Cragin	10.8	18.7	14.6	37.3	37.3	154
19	20.0	Hermosa	6.9	20.5	13.1	41.6	36.4	150
20	21.0	Avondale	6.0	15.3	9.2	24.7	31.0	200
21	22.0	Logan Square	3.2	16.8	8.2	14.8	26.2	319
22	23.0	Humboldt park	14.8	33.9	17.3	35.4	38.0	137
23	24.0	West Town	2.3	14.7	6.6	12.9	21.7	431
24	25.0	Austin	6.3	28.6	22.6	24.4	37.9	159
25	26.0	West Garfield Park	9.4	41.7	25.8	24.5	43.6	109
26	27.0	East Garfield	8.2	42.4	19.6	21.3	43.2	129

In [204]:

```
crimes1['crimes_community']=community_crime_count
```

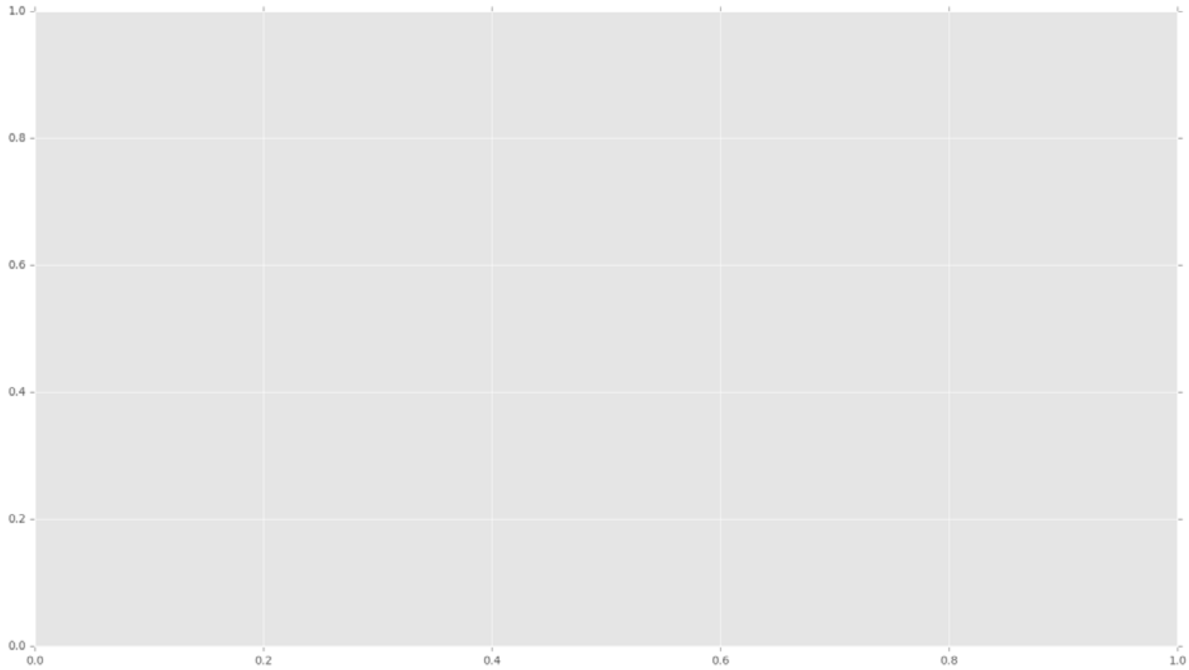
```
In [207]: crimes1
```

```
Out[207]:
```

	Community Area Number	COMMUNITY AREA NAME	PERCENT OF HOUSING CROWDED	PERCENT HOUSEHOLDS BELOW POVERTY	PERCENT AGED 16+ UNEMPLOYED	PERCENT AGED 25+ WITHOUT HIGH SCHOOL DIPLOMA	PERCENT AGED UNDER 18 OR OVER 64	PER
0	1.0	Rogers Park	7.7	23.6	8.7	18.2	27.5	2393
1	2.0	West Ridge	7.8	17.2	8.8	20.8	38.5	2304
2	3.0	Uptown	3.8	24.0	8.9	11.8	22.2	3578
3	4.0	Lincoln Square	3.4	10.9	8.2	13.4	25.5	3752
4	5.0	North Center	0.3	7.5	5.2	4.5	26.2	5712
5	6.0	Lake View	1.1	11.4	4.7	2.6	17.0	6005
6	7.0	Lincoln Park	0.8	12.3	5.1	3.6	21.5	7155
7	8.0	Near North Side	1.9	12.9	7.0	2.5	22.6	8866
8	9.0	Edison Park	1.1	3.3	6.5	7.4	35.3	4095
9	10.0	Norwood Park	2.0	5.4	9.0	11.5	39.5	3287
10	11.0	Jefferson Park	2.7	8.6	12.4	13.4	35.5	2775
11	12.0	Forest Glen	1.1	7.5	6.8	4.9	40.5	4416
12	13.0	North Park	3.9	13.2	9.9	14.4	39.0	2657
13	14.0	Albany Park	11.3	19.2	10.0	32.9	32.0	2132
14	15.0	Portage Park	4.1	11.6	12.6	19.3	34.0	2433
15	16.0	Irving Park	6.3	13.1	10.0	22.4	31.6	2724
16	17.0	Dunning	5.2	10.6	10.0	16.2	33.6	2628
17	18.0	Montclaire	8.1	15.3	13.8	23.5	38.6	220
18	19.0	Belmont Cragin	10.8	18.7	14.6	37.3	37.3	1546
19	20.0	Hermosa	6.9	20.5	13.1	41.6	36.4	1508
20	21.0	Avondale	6.0	15.3	9.2	24.7	31.0	2003
21	22.0	Logan Square	3.2	16.8	8.2	14.8	26.2	3190
22	23.0	Humboldt park	14.8	33.9	17.3	35.4	38.0	1378
23	24.0	West Town	2.3	14.7	6.6	12.9	21.7	4319
24	25.0	Austin	6.3	28.6	22.6	24.4	37.9	1595
25	26.0	West Garfield Park	9.4	41.7	25.8	24.5	43.6	1093
26	27.0	East Garfield	8.2	42.4	19.6	21.3	43.2	1296

In [215]:
```
crimes1.plot.scatter(x="crimes_communitya",y="PER_CAPITA_INCOME")
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
C:\Users\PC\Anaconda3\lib\site-packages\pandas\indexes\base.py in get_loc(self,
key, method, tolerance)
   1944             try:
-> 1945                 return self._engine.get_loc(key)
   1946             except KeyError:

pandas\index.pyx in pandas.index.IndexEngine.get_loc (pandas\index.c:4154)()

pandas\index.pyx in pandas.index.IndexEngine.get_loc (pandas\index.c:4018)()

pandas\hashtable.pyx in pandas.hashtable.PyObjectHashTable.get_item (pandas\hash
table.c:12368)()

pandas\hashtable.pyx in pandas.hashtable.PyObjectHashTable.get_item (pandas\hash
table.c:12322)()

KeyError: 'crimes_communitya'

During handling of the above exception, another exception occurred:

KeyError                                  Traceback (most recent call last)
<ipython-input-215-e1373b80cd25> in <module>()
----> 1 crimes1.plot.scatter(x="crimes_communitya",y="PER_CAPITA_INCOME")

C:\Users\PC\Anaconda3\lib\site-packages\pandas\tools\plotting.py in scatter(self
, x, y, s, c, **kwds)
   3916         axes : matplotlib.AxesSubplot or np.array of them
   3917         """
-> 3918         return self(kind='scatter', x=x, y=y, c=c, s=s, **kwds)
   3919 
   3920     def hexbin(self, x, y, C=None, reduce_C_function=None, gridsize=None
,

C:\Users\PC\Anaconda3\lib\site-packages\pandas\tools\plotting.py in __call__(sel
f, x, y, kind, ax, subplots, sharex, sharey, layout, figsize, use_index, title,
grid, legend, style, logx, logy, loglog, xticks, yticks, xlim, ylim, rot, fontsi
ze, colormap, table, yerr, xerr, secondary_y, sort_columns, **kwds)
   3738                           fontsize=fontsize, colormap=colormap, table=ta
ble,
   3739                           yerr=yerr, xerr=xerr, secondary_y=secondary_y,
-> 3740                           sort_columns=sort_columns, **kwds)
   3741     __call__.__doc__ = plot_frame.__doc__
   3742 

C:\Users\PC\Anaconda3\lib\site-packages\pandas\tools\plotting.py in plot_frame(d
ata, x, y, kind, ax, subplots, sharex, sharey, layout, figsize, use_index, title
, grid, legend, style, logx, logy, loglog, xticks, yticks, xlim, ylim, rot, font
size, colormap, table, yerr, xerr, secondary_y, sort_columns, **kwds)
   2612                  yerr=yerr, xerr=xerr,
   2613                  secondary_y=secondary_y, sort_columns=sort_columns,
-> 2614                  **kwds)
   2615 
   2616 

C:\Users\PC\Anaconda3\lib\site-packages\pandas\tools\plotting.py in _plot(data,
x, y, subplots, ax, kind, **kwds)
   2439         plot_obj = klass(data, subplots=subplots, ax=ax, kind=kind, **kw
ds)
   2440 
-> 2441     plot_obj.generate()
   2442     plot_obj.draw()
   2443     return plot_obj.result
```


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0