Taller 7

Métodos Computacionales para Políticas Públicas - URosario

Entrega: viernes 11-oct-2019 11:59 PM

Juan Sebastián Muñoz

jsebastianmvargas@gmail.com (mailto:jsebastianmvargas@gmail.com)

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/data (https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present/ijzp-q8t2/data)).

Parte obligatoria

In [17]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
plt.rcParams["figure.figsize"] = [28.0, 20.0]
plt.style.use('ggplot')
```

1.

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

In [8]:
```
crimes = pd.read_csv('Crimes_-_2001_to_present.csv', parse_dates=['Date'])
```

In [194]:
```
crimes.head()
```

Out[194]:

	ID	Case Number	Date	Block	IUCR	Primary Type	Description	Location Description	Arr
0	10201852	HY389096	2015-01-01	008XX N MAPLEWOOD AVE	2825	OTHER OFFENSE	HARASSMENT BY TELEPHONE	APARTMENT	Fa
1	10060114	HY239140	2015-01-01	069XX S CORNELL AVE	1751	OFFENSE INVOLVING CHILDREN	CRIM SEX ABUSE BY FAM MEMBER	RESIDENCE	Fa
2	10210454	HY397301	2015-01-01	049XX W WABANSIA AVE	0266	CRIM SEXUAL ASSAULT	PREDATORY	RESIDENCE	Fa
3	10025440	HY214766	2015-01-01	004XX E 80TH ST	1154	DECEPTIVE PRACTICE	FINANCIAL IDENTITY THEFT $300 AND UNDER	RESIDENCE	Fa
4	10225520	HY412735	2015-01-01	075XX S BLACKSTONE AVE	1153	DECEPTIVE PRACTICE	FINANCIAL IDENTITY THEFT OVER $ 300	RESIDENCE	Fa

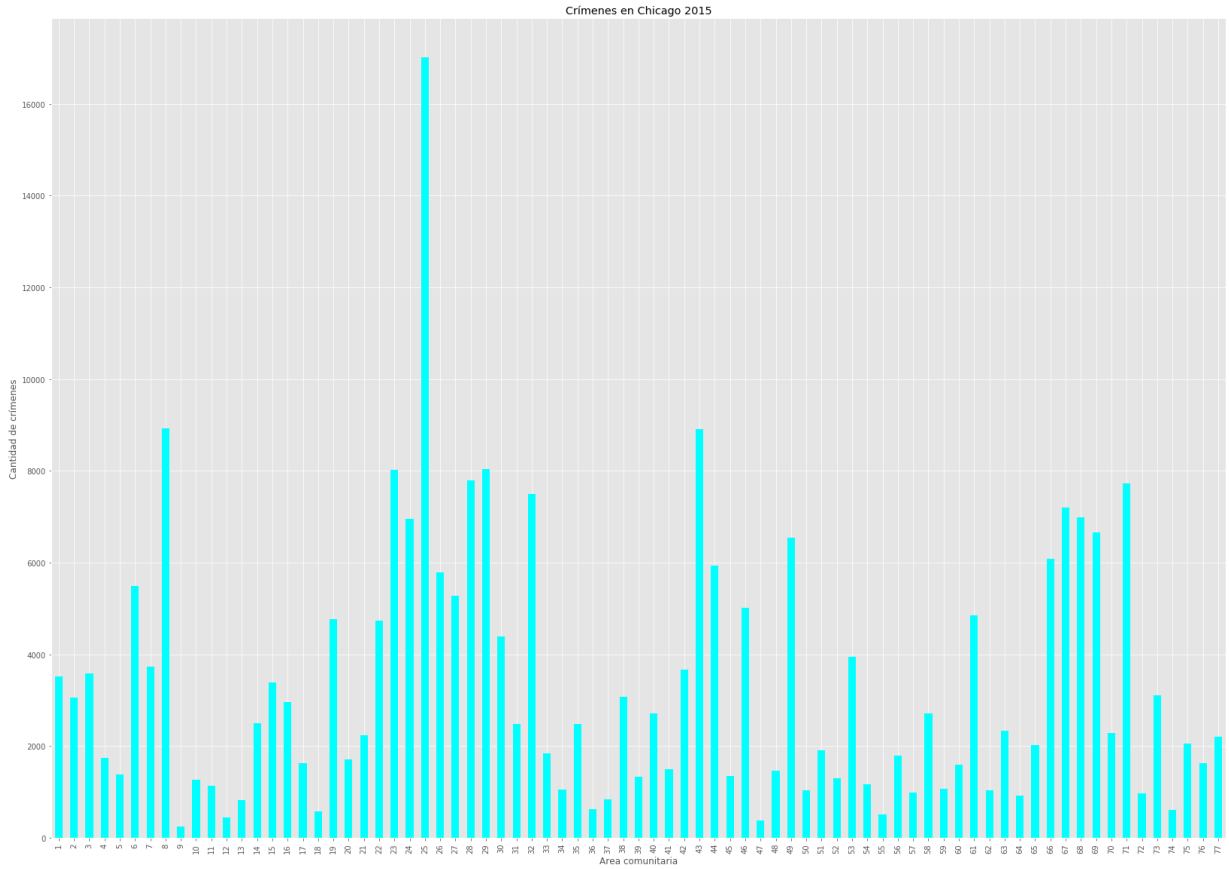
5 rows × 23 columns

In [10]:
```
#Agrupación de los casos por Area comunitaria
crimes_by_community = crimes.groupby("Community Area")
```

In [195]:
```
#Conteo de los crimenes por Area comunitaria
community_crime_count = crimes_by_community['ID'].agg('count')
community_crime_count.head()
```

Out[195]:
```
Community Area
1    3519
2    3059
3    3585
4    1747
5    1375
Name: ID, dtype: int64
```

In [20]:
```
#Grafico de los crímenes por area comunitaria en 2015
community_crime_count.plot(kind='bar', color ="cyan")
plt.title("Crímenes en Chicago 2015")
plt.xlabel("Area comunitaria")
plt.ylabel("Cantidad de crímenes");
```



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 [22]:
```
type(community_crime_count)
```

Out[22]:
```
pandas.core.series.Series
```

In [27]:
```
community_crime_count.sort_values(ascending=True).head(5)
```

Out[27]:
```
Community Area
9     254
47    380
12    444
55    506
18    572
Name: ID, dtype: int64
```

'Edison Park' es la Community Area con menos crímenes en Chicago para 2015

In [28]:
```
community_crime_count.sort_values(ascending=False).head(5)
```

Out[28]:
```
Community Area
25    17020
8      8920
43     8906
29     8039
23     8015
Name: ID, dtype: int64
```

'Austin' es la Community Area con más crímenes en Chicago para 2015

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.

In [31]:
```
# Create function to strip time from date field, and use it to create another co
def to_day(timestamp):
    return timestamp.replace(minute=0,hour=0, second=0)

crimes['Day'] = crimes['Date'].apply(to_day)
```

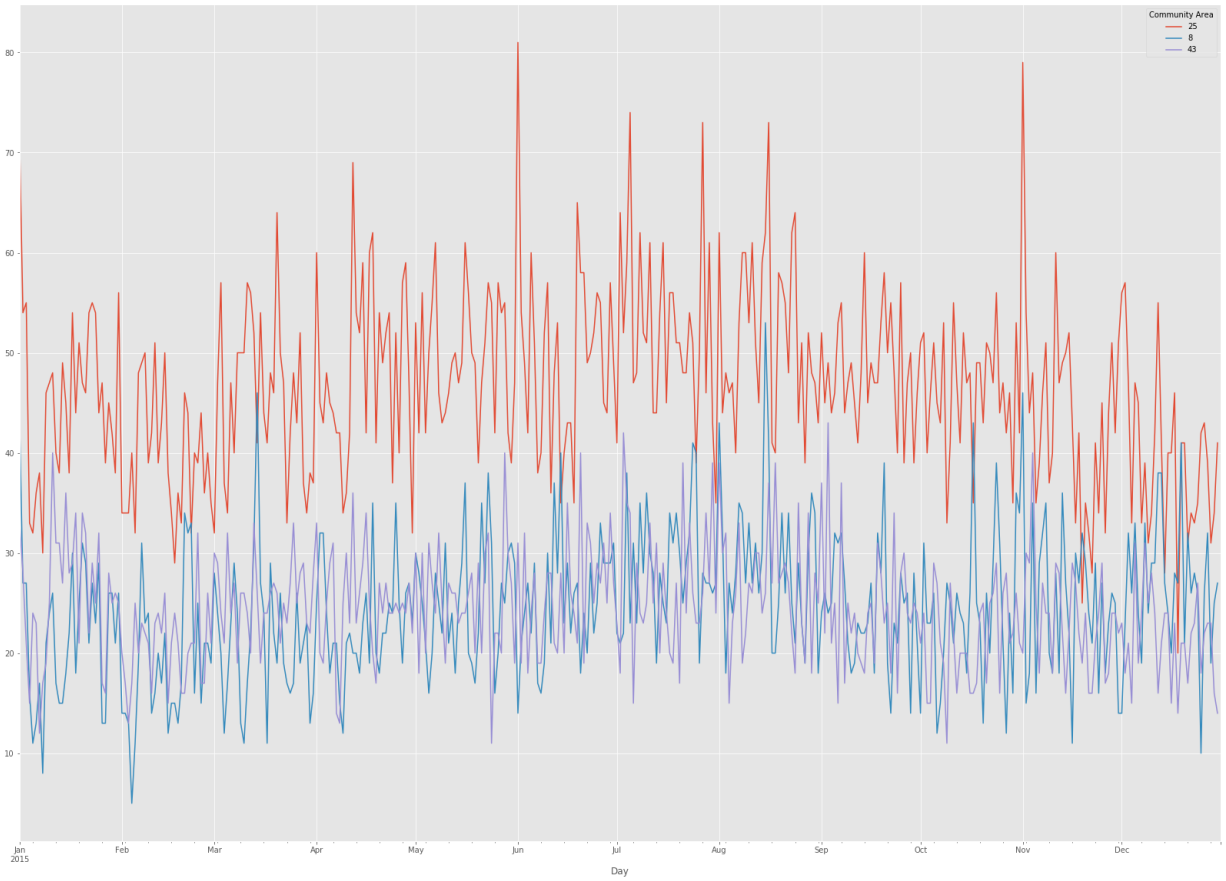
In [196]:
```
#Tabla de crímenes por Community Area cada día del año
crimes_by_community_day = crimes.groupby(['Community Area', 'Day'])
crimes_by_community_day_count = crimes_by_community_day['ID'].agg('count')
crimes_community_day_timeseries = crimes_by_community_day_count.unstack('Communi
crimes_community_day_timeseries.head()
```

Out[196]:

Community Area	1	2	3	4	5	6	7	8	9	10	...	68	69	70	71	72
Day																
2015-01-01	13.0	7.0	11.0	4.0	5.0	22.0	12.0	43.0	1.0	5.0	...	29.0	23.0	9.0	44.0	2.0
2015-01-02	5.0	9.0	8.0	3.0	2.0	10.0	9.0	27.0	NaN	2.0	...	12.0	21.0	5.0	17.0	1.0
2015-01-03	7.0	11.0	9.0	7.0	4.0	6.0	11.0	27.0	1.0	3.0	...	23.0	12.0	8.0	18.0	NaN
2015-01-04	12.0	7.0	9.0	10.0	3.0	15.0	5.0	16.0	1.0	4.0	...	13.0	15.0	9.0	12.0	1.0
2015-01-05	6.0	7.0	5.0	4.0	5.0	15.0	7.0	11.0	1.0	3.0	...	16.0	12.0	8.0	17.0	NaN

5 rows × 77 columns

In [45]:
```
#Serie de tiempo de las tres Community areas con más crímenes en todo 2015
crimes_community_day_timeseries[[25, 8, 43]].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)).

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 [197]:
```
community_crime_count = community_crime_count.to_frame()
```

In [198]:
```
community_crime_count = community_crime_count.reset_index(level=['Community Area
```

In [206]:
```
socio_economic = pd.read_csv("Census_Data_-_Selected_socioeconomic_indicators_in_
```

In [207]:
```
socio_economic.head()
```

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	CA INC
0	1.0	Rogers Park	7.7	23.6	8.7	18.2	27.5	2
1	2.0	West Ridge	7.8	17.2	8.8	20.8	38.5	2
2	3.0	Uptown	3.8	24.0	8.9	11.8	22.2	3
3	4.0	Lincoln Square	3.4	10.9	8.2	13.4	25.5	3
4	5.0	North Center	0.3	7.5	5.2	4.5	26.2	5

In [208]:
```
socio_economic = socio_economic.drop(77)
```

In [209]:
```
socio_economic.insert(0, 'Community_Area', socio_economic['Community Area Number
socio_economic = socio_economic.drop(columns="Community Area Number")
socio_economic.head()
```

Out[209]:

	Community_Area	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
0	1	Rogers Park	7.7	23.6	8.7	18.2	27.5
1	2	West Ridge	7.8	17.2	8.8	20.8	38.5
2	3	Uptown	3.8	24.0	8.9	11.8	22.2
3	4	Lincoln Square	3.4	10.9	8.2	13.4	25.5
4	5	North Center	0.3	7.5	5.2	4.5	26.2

In [190]:
```
community_crime_count.columns = ["Community_Area","Crimes"]
socio_economic.columns = ['Community_Area', 'COMMUNITY AREA NAME', 'PERCENT OF HO
       'PERCENT HOUSEHOLDS BELOW POVERTY', 'PERCENT AGED 16+ UNEMPLOYED',
       'PERCENT AGED 25+ WITHOUT HIGH SCHOOL DIPLOMA',
       'PERCENT AGED UNDER 18 OR OVER 64', 'Per_Capita_Income',
       'HARDSHIP INDEX']
```

In [191]:
```
crimes_socioeconomic = pd.merge(community_crime_count, socio_economic, on='Commu
```

In [167]:
```
## No entiendo porque me dice que no tengo los datos de crimenes si arriba aparec
## Intento hacer el gráfico con otras columnas y me deja, pero con esa no.
Scatter_plot = socio_economic.plot.scatter(x='crimes', y='Per_Capita_Income', c=
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
~\Anaconda3\lib\site-packages\pandas\core\indexes\base.py in get_loc(self, key,
method, tolerance)
   2656             try:
-> 2657                 return self._engine.get_loc(key)
   2658             except KeyError:

pandas/_libs/index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas/_libs/index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashT
able.get_item()

pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashT
able.get_item()

KeyError: 'crimes'

During handling of the above exception, another exception occurred:

KeyError                                  Traceback (most recent call last)
<ipython-input-167-8967ca82034c> in <module>
----> 1 aa = socio_economic.plot.scatter(x='crimes', y='Per_Capita_Income', c=
'DarkBlue')

~\Anaconda3\lib\site-packages\pandas\plotting\_core.py in scatter(self, x, y,
 s, c, **kwds)
   3514             ...                       colormap='viridis')
   3515         """
-> 3516         return self(kind='scatter', x=x, y=y, c=c, s=s, **kwds)
   3517
   3518     def hexbin(self, x, y, C=None, reduce_C_function=None, gridsize=Non
e,

~\Anaconda3\lib\site-packages\pandas\plotting\_core.py in __call__(self, x, y,
 kind, ax, subplots, sharex, sharey, layout, figsize, use_index, title, grid, l
egend, style, logx, logy, loglog, xticks, yticks, xlim, ylim, rot, fontsize, co
lormap, table, yerr, xerr, secondary_y, sort_columns, **kwds)
   2940                           fontsize=fontsize, colormap=colormap, table=t
able,
   2941                           yerr=yerr, xerr=xerr, secondary_y=secondary_y
,
-> 2942                           sort_columns=sort_columns, **kwds)
   2943     __call__.__doc__ = plot_frame.__doc__
   2944

~\Anaconda3\lib\site-packages\pandas\plotting\_core.py in plot_frame(data, x,
 y, kind, ax, subplots, sharex, sharey, layout, figsize, use_index, title, gri
d, legend, style, logx, logy, loglog, xticks, yticks, xlim, ylim, rot, fontsiz
e, colormap, table, yerr, xerr, secondary_y, sort_columns, **kwds)
   1971                  yerr=yerr, xerr=xerr,
```

```
   1972                 secondary_y=secondary_y, sort_columns=sort_columns,
-> 1973                 **kwds)
   1974
   1975

~\Anaconda3\lib\site-packages\pandas\plotting\_core.py in _plot(data, x, y, sub
plots, ax, kind, **kwds)
   1738         if isinstance(data, ABCDataFrame):
   1739             plot_obj = klass(data, x=x, y=y, subplots=subplots, ax=ax,
-> 1740                              kind=kind, **kwds)
   1741         else:
   1742             raise ValueError("plot kind %r can only be used for data fr
ames"

~\Anaconda3\lib\site-packages\pandas\plotting\_core.py in __init__(self, data,
 x, y, s, c, **kwargs)
    858             # the handling of this argument later
    859             s = 20
--> 860         super(ScatterPlot, self).__init__(data, x, y, s=s, **kwargs)
    861         if is_integer(c) and not self.data.columns.holds_integer():
    862             c = self.data.columns[c]

~\Anaconda3\lib\site-packages\pandas\plotting\_core.py in __init__(self, data,
 x, y, **kwargs)
    801         if is_integer(y) and not self.data.columns.holds_integer():
    802             y = self.data.columns[y]
--> 803         if len(self.data[x]._get_numeric_data()) == 0:
    804             raise ValueError(self._kind + ' requires x column to be num
eric')
    805         if len(self.data[y]._get_numeric_data()) == 0:

~\Anaconda3\lib\site-packages\pandas\core\frame.py in __getitem__(self, key)
   2925             if self.columns.nlevels > 1:
   2926                 return self._getitem_multilevel(key)
-> 2927             indexer = self.columns.get_loc(key)
   2928             if is_integer(indexer):
   2929                 indexer = [indexer]

~\Anaconda3\lib\site-packages\pandas\core\indexes\base.py in get_loc(self, key,
method, tolerance)
   2657                 return self._engine.get_loc(key)
   2658             except KeyError:
-> 2659                 return self._engine.get_loc(self._maybe_cast_indexer(ke
y))
   2660         indexer = self.get_indexer([key], method=method, tolerance=tole
rance)
   2661         if indexer.ndim > 1 or indexer.size > 1:

pandas/_libs/index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas/_libs/index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashT
able.get_item()

pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashT
able.get_item()
```

```
KeyError: 'crimes'
```

```
---
```