

mcpp_taller7_julian_ramirez

October 11, 2019

1 Taller 7

Métodos Computacionales para Políticas Públicas - UROSARIO

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

Julián Santiago Ramírez

julians.ramirez@urosario.edu.co

1.1 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_mataallana
- 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>).

1.1.1 Parte obligatoria

1.1.2 1.

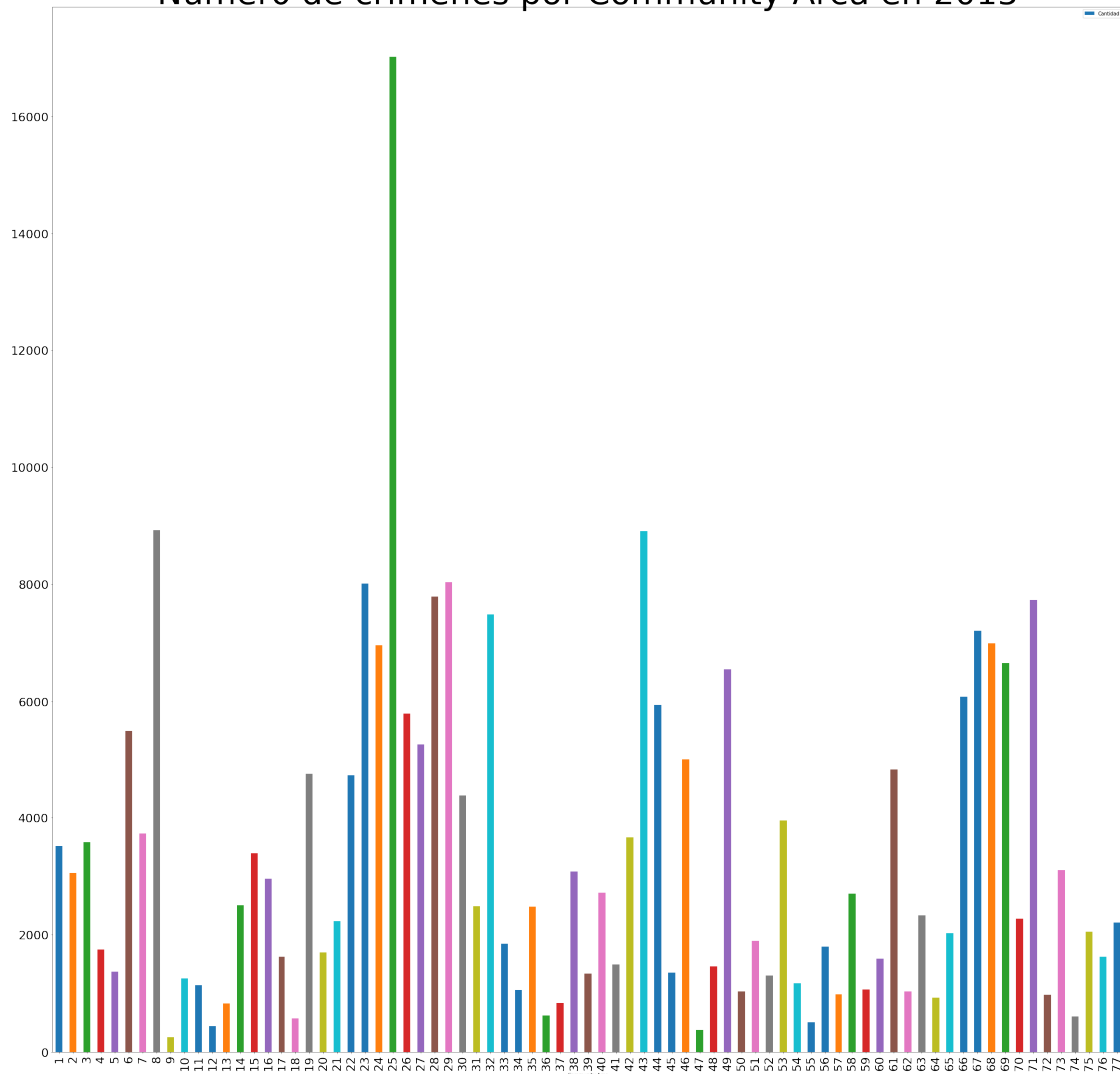
Calcule el número de crímenes en cada Community Area en 2015. Haga un gráfico de barras que lo ilustre.

```
In [1]: import pandas as pd
import matplotlib.pyplot as plt

In [14]: ### Guardamos el csv en una variable que se llama tabla
tabla = pd.read_csv('Crimes_-_2001_to_present.csv')
### Ahora tenemos que agrupar los datos por Community Area
Agrupados = tabla.groupby(['Community Area']).size().reset_index(name='Cantidad')
### Finalmente procedemos a crear el gráfico de barras
grafica= Agrupados.plot.bar(x='Community Area', y='Cantidad',fontSize=25)
plt.rcParams["figure.figsize"] = [40.0, 40.0]
grafica.set_title("Número de crímenes por Community Area en 2015",fontSize=70)
grafica

Out[14]: <matplotlib.axes._subplots.AxesSubplot at 0x1f786f53630>
```

Número de crímenes por Community Area en 2015



1.1.3 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 [80]: ### Organizamos la variable Agrupados por el valor de "Cantidad" de menor a mayor
Agrupados.sort_values(by=['Cantidad'], inplace=True)
### Community area con menor numero de crímenes
menor=Agrupados["Community Area"].iloc[0]
y=len(Agrupados)
mayor=Agrupados["Community Area"].iloc[y-1]
```

```

print("El área que presenta menor número de crímenes es la: ",menor)
print("El área que presenta mayor número de crímenes es la: ",mayor)
##

```

El área que presenta menor número de crímenes es la: 9
El área que presenta mayor número de crímenes es la: 25

1.1.4 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.

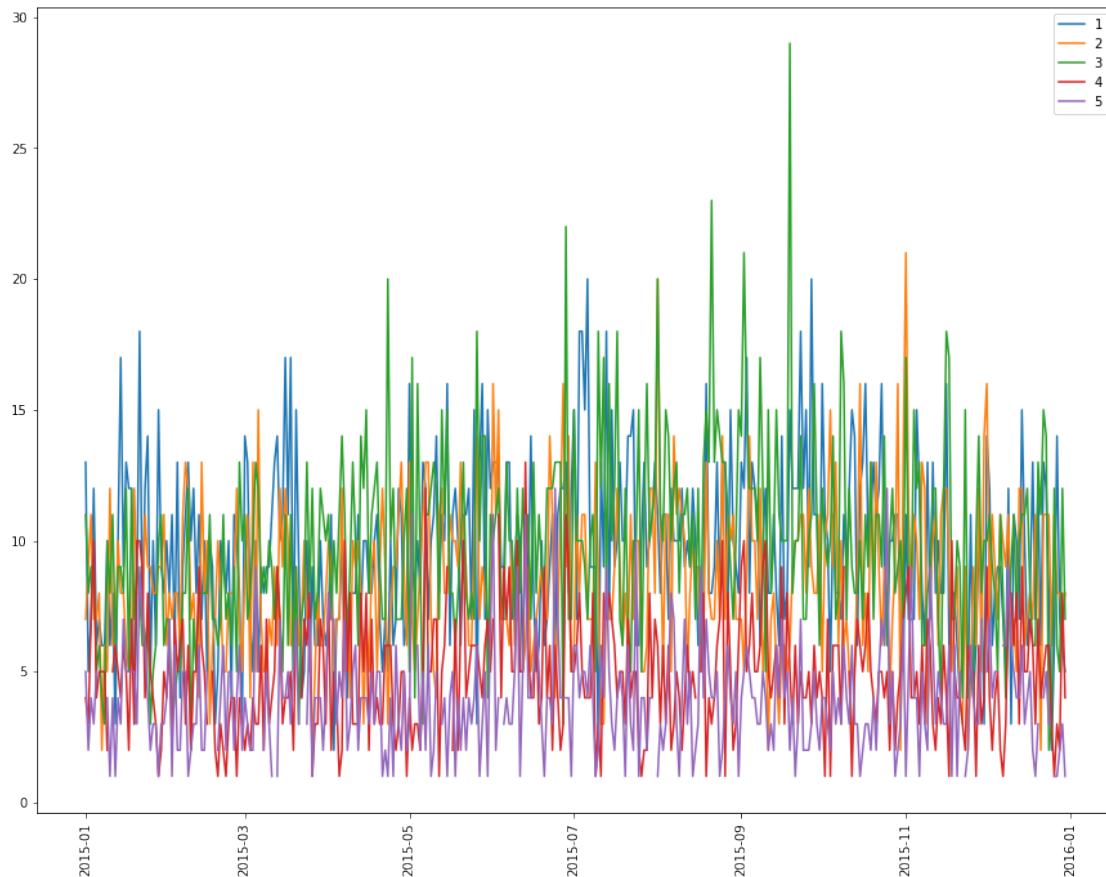
```

In [86]: # 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)

In [91]: ##### No me funciona aplicando la funcion "to_day"
#tabla['Fecha'] = tabla['Date'].apply(to_day)
##### Aplico entonces esta solución
tabla['Fecha'] = pd.to_datetime(tabla['Date']).dt.date

In [102]: ### crearemos la tabla con la fecha y las communities areas
total=tabla[['Fecha','Community Area']]
### Contamos la cantidad de crímenes
total = total.groupby(['Fecha','Community Area']).size().reset_index(name='cantidad')
### Community areas interesantes
total = total[(total['Community Area'] < 6)]
### Para graficar
total_grafico = total.pivot_table('cantidad',['Fecha'],'Community Area')
### Graficamos
fig, ax = plt.subplots()
fig.subplots_adjust(bottom=0.3)
plt.xticks(rotation=90)
plt.plot(total_grafico)
plt.legend(('1','2','3','4','5'))
plt.rcParams["figure.figsize"] = [15.0, 15.0]

```



1.1.5 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>).

1.1.6 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.
