

Alba Santos Rodríguez alu0100953657@ull.edu.es

▼ Ejercicios teóricos

Ejercicio 1

¿Qué consigues con groupby?

Se usa para agrupar los datos de un data frame por la fila que queramos.

Ejercicio 2

¿Por qué es importante una buena visualización?

Porque facilita el análisis posterior de los datos

Ejercicio 3

¿Con qué línea de código muestro una gráfica de barras? ¿Y una de queso?

`plt.bar()` `plt.pie()`

Ejercicio 4

¿Cómo hacer un bot de Telegram? Haz una lista de pasos generales, no quiero que me hables de código.

Primero instalamos Telegram y creamos una conversación con BotFather, donde escribimos `\start`. A continuación le decimos que queremos crear un nuevo bot con `\newbot` y le decimos el nombre y el nombre de usuario de nuestro bot. Y a partir de ahí podremos programarlo con el Token que nos da BotFather

▼ Ejercicios prácticos

▼ Ejercicio 8 del día 3

Agrupar por media por year, month y esta vez también day. Una vez hecho eso muestra la media de generación eólica para el día 3 de junio de 2015

Resolver

```
media_por_año = datos.groupby('year').mean()
```

```
media_por_ano = datos.groupby('year').mean()  
media_por_mes = datos.groupby('month').mean()  
media_por_day = datos.groupby('day').mean()  
  
datos[(datos['day'] == 3)&(datos['month'] == 6)&(datos['year'] == 2015)].mean()
```

▼ Ejercicio 1

Visualizar la evolución no de los casos totales sino de los casos nuevos de España mediante un gráfico de líneas.

Resolver

```
import matplotlib.pyplot as plt  
import matplotlib as mpl  
  
mpl.rcParams['font.size'] = 15.0  
datos_españa = datos[datos['location'] == 'Spain']  
x = datos_españa['date']  
y = datos_españa['new_cases']  
  
fig = plt.figure()  
ax = fig.add_axes([1, 1, 2, 2]) # main axes  
  
ax.set_xlabel('Fechas') # nombre eje x  
ax.set_ylabel('Cantidad de casos') # nombre eje y  
ax.set_title('Nuevos casos en España') # nombre gráfica  
  
ax.set_yticks([0,500000,1000000,1500000])  
ax.set_yticklabels(['0','500 mil','1 mill', '1.5 mill'])  
  
plt.plot(x, y, lw=3)  
plt.show()
```

▼ Ejercicio 2

Ahora hacer lo mismo pero para los casos nuevos y no los totales, con todos los países

▼ Ejercicio 3

Ahora hacer lo mismo pero para los casos nuevos con una gráfica de barras pero con Estados Unidos en vez de España

Resolver

```
import matplotlib.pyplot as plt
import matplotlib as mpl

mpl.rcParams['font.size'] = 30.0
datos_estadosunidos = datos[datos['location'] == 'United States']
x = datos_estadosunidos['date']
y = datos_estadosunidos['new_cases']

fig = plt.figure()
ax = fig.add_axes([1, 1, 2, 2]) # main axes

ax.set_xlabel('Cantidad') # nombre eje x
ax.set_ylabel('Frecuencia') # nombre eje y
ax.set_title('Frecuencia de nuevos casos en Estados Unidos') # nombre gráfica

ax.set_xticklabels(['feb', 'mar', 'abr', 'may', 'jun',
                    'jul', 'ago', 'sep', 'oct', 'nov', 'dic'])

plt.bar(x, y)
plt.show()
```

▼ Ejercicio 4

Ahora hacer una gráfica de queso para los casos totales sólo de America del Sur.

```
## Resolver
import matplotlib.pyplot as plt
import matplotlib as mpl
mpl.rcParams['font.size'] = 30.0

datos_sur = datos[datos['continent']=='South America']

values = datos_sur.total_cases
print(len(datos_sur.total_cases))
colors = ['r', 'g', 'b', 'm', 'c', '#B3AA19', 'r', 'g', 'b', 'm', 'c', '#B3AA19']
print(len(colors))

labels = datos_sur['location']
plt.pie(values, colors= colors, labels=labels)
fig = plt.gcf()
fig.set_size_inches(15,15) # Para cambiar el tamaño

plt.title('Reparto de casos totales en sudamérica')
plt.show()
fig.savefig("reparto_queso.png") # Calidad resolucion
```

▼ Ejercicio 5

Queremos lo mismo de la gráfica de puntos que vimos pero queremos ver la evolución de casos totales con puntos de junio, julio y agosto

```
## Resolver
```

▼ Ejercicio 6

Añadir a la esta gráfica que acabamos de ver de Plotly: Reino Unido, Francia, Alemania, Dinamarca y Bélgica

```
## Resolver
```

```
import plotly.graph_objs as go
from plotly.offline import init_notebook_mode, iplot
import plotly as ply
from plotly import subplots

datos_españa = datos[datos['location'] == 'Spain']
datos_italia = datos[datos['location'] == 'Italy']
datos_reinounido = datos[datos['location'] == 'United Kingdom']
datos_francia = datos[datos['location'] == 'France']
datos_dinamarca = datos[datos['location'] == 'Denmark']
datos_alemania = datos[datos['location'] == 'Germany']
datos_belgica = datos[datos['location'] == 'Belgium']

x_es = datos_españa['date']
y_es = datos_españa['total_cases']

x_it = datos_italia['date']
y_it = datos_italia['total_cases']

x_den = datos_dinamarca['date']
y_den = datos_dinamarca['total_cases']

x_ru = datos_reinounido['date']
y_ru = datos_reinounido['total_cases']

x_fr = datos_francia['date']
y_fr = datos_francia['total_cases']

x_al = datos_alemania['date']
y_al = datos_alemania['total_cases']

x_be = datos_belgica['date']
y_be = datos_belgica['total_cases']
```

```
# Create traces
```

```
trace0 = go.Scatter(
```

```
trace0 = go.Scatter(
    x = x_es,
    y = y_es,
    mode = 'lines',
    name = 'Total casos España'
)

trace1 = go.Scatter(
    x = x_it,
    y = y_it,
    mode = 'lines',
    name = 'Total casos Italia'
)

trace2 = go.Scatter(
    x = x_den,
    y = y_den,
    mode = 'lines',
    name = 'Total casos Dinamarca'
)

trace3 = go.Scatter(
    x = x_ru,
    y = y_ru,
    mode = 'lines',
    name = 'Total casos Reino Unido'
)

trace4 = go.Scatter(
    x = x_fr,
    y = y_fr,
    mode = 'lines',
    name = 'Total casos Francia'
)

trace5 = go.Scatter(
    x = x_al,
    y = y_al,
    mode = 'lines',
    name = 'Total casos Alemania'
)

trace6 = go.Scatter(
    x = x_be,
    y = y_be,
    mode = 'lines',
    name = 'Total casos Bélgica'
)

data = [trace0, trace1, trace2, trace3, trace4, trace5, trace6]

iplot(data)
```

