

## Índice de contenidos

<b>Objetivos.</b>	<b>2</b>
<b>Actividades a desarrollar</b>	<b>3</b>
<b>Desarrollo de las actividades</b>	<b>3</b>
Origen de los datos	3
Estrategia de programación	3
Librerías usadas en el script	3
Carga de datos en un dataframe, a partir del csv	4
Obtención de número de estaciones con "total" de 30	4
Número de estación con la media más alta de bicis disponibles	5
Histograma de la estación de bicis disponibles	5
Gráfica con la línea temporal de bicis disponibles	6
Referencias usadas	6
<b>Resultados de las actividades</b>	<b>7</b>
Carga de datos en un dataframe, a partir del csv	7
Obtención de número de estaciones con "total" de 30	7
Número de estación con la media más alta de bicis disponibles	7
Histograma de la estación de bicis disponibles	8
Gráfica con la línea temporal de bicis disponibles	8
<b>Conclusiones</b>	<b>9</b>

## 1. Objetivos.

- Realizar consultas sobre un entorno de programación en Python para analizar datos
- Aplicar técnicas de tratamiento exploratorio de datos y realizar visualizaciones
- Entender y comprender el uso de los entornos de programación y sus objetivos
- Entender las diferentes posibilidades del uso de Pandas

## 2. Actividades a desarrollar

- Carga de datos de csv (estaciones\_bici.csv) en un DataFrame
- Obtención de número de estaciones con total de 30
- Número de estación con la media más alta de bicis disponibles
  - Partiendo de esta estación:
    - Realizar el histograma de la estación de bicis disponibles
    - Realizar gráfica con la línea temporal de bicis disponibles

## 3. Desarrollo de las actividades

### 3.1. Origen de los datos

El *dataset* está almacenado en formato csv en el mismo directorio que el script Notebook, se trata de un fichero llamado **estaciones\_bici.csv** que contiene datos sobre el servicio de [ValenBisi](#) del Ayuntamiento de Valencia.

### 3.2. Estrategia de programación

Librerías usadas en el script

```
# Importación de librerías
```

```
import numpy as np
```

```
import pandas as pd
```

```
import matplotlib.pyplot as plt

# la siguiente línea permitirá mostrar el gráfico sin necesidad de
    llamar luego a plt.show()

%matplotlib inline
```

## Carga de datos en un dataframe, a partir del csv

```
df_estaciones = pd.read_csv('./estaciones_bici.csv', delimiter=';')
df_estaciones.describe() # muestra estadísticas del dataframe
```

### Código extra:

```
df_estaciones # muestra todas las filas del dataframe (en realidad solo
    las 5 primeras y 5 últimas)

print(len(df_estaciones)) # muestra cuántas estaciones hay en total

df_estaciones.head() # muestra solo las 5 primeras filas del dataframe
```

## Obtención de número de estaciones con "total" de 30

```
df_estaciones.query('total==30').total.count()
```

### Código extra:

```
consulta1 = df_estaciones[df_estaciones['total']==30]

print(consulta1) # muestra las estaciones que cumplen la condición

print(len(consulta1)) # muestra cuántas estaciones cumplen la condición
```

## Número de estación con la media más alta de bicis disponibles

```
ls_estaciones = df_estaciones.groupby('estacion')['available'].mean() #  
    obtenemos lista con la media por cada estación  
df2 = pd.DataFrame(ls_estaciones) # convierte el resultado anterior a  
    DataFrame usando la librería Pandas  
df2 = df2.rename(columns={"available": "average_available"}) #  
    renombramos columna 'available' a 'average_available'  
df2.average_available = df2.average_available.round(2) # modificamos la  
    columna 'average_available' para tener solo 2 decimales  
df2 = df2.sort_values('average_available', ascending=False) # ordenamos  
    las filas descendientemente por average_available  
df2 = df2.head(1) # nos quedamos con la primera: la más alta  
df2 # muestra la estación y su media, que es la más alta de todas.  
  
# con lo anterior sería suficiente pero vamos a guardar el número de  
    estación en una variable  
estacion = df2.index[0] #'station' forma parte del índice, el primer  
    valor que retorna el método index es el número del índice  
estacion
```

## Histograma de la estación de bicis disponibles

```
df_estaciones.query('estacion==@estacion')  
plt.hist(df_estaciones["available"])
```

## Gráfica con la línea temporal de bicis disponibles

```
# Nos quedamos solo con los de la estación anterior
df3 = df_estaciones.query('estacion==@estacion')

# A partir del dataframe anterior nos quedamos solo con las columnas
  'updated' y 'available'
df3 = df3.loc[:, ['updated', 'available']]

# Ordenar la serie temporal
df3 = df3.sort_values('updated', ascending=True)

# Plotear la serie
plt.plot(df3['updated'], df3['available'])
plt.xticks(rotation='vertical')
```

## Referencias usadas

1. [pandas.pydata.org/pandas-docs/stable/reference/index.html](https://pandas.pydata.org/pandas-docs/stable/reference/index.html) [consulta: 14/03/2020]
2. <https://numpy.org/> [consulta: 14/03/2020]
3. <https://matplotlib.org/> [consulta: 14/03/2020]

## 4. Resultados de las actividades

Carga de datos en un dataframe, a partir del csv

	Unnamed: 0	available	connected	estation	free	open	ticket	total
count	27547.000000	27547.000000	27547.000000	27547.000000	27547.000000	27547.0	27547.000000	27547.000000
mean	13773.000000	8.974444	0.996261	138.449196	10.629397	1.0	0.503794	19.915381
std	7952.278269	7.307137	0.061035	79.657747	7.492671	0.0	0.499995	5.570912
min	0.000000	0.000000	0.000000	1.000000	0.000000	1.0	0.000000	10.000000
25%	6886.500000	3.000000	1.000000	69.000000	4.000000	1.0	0.000000	15.000000
50%	13773.000000	8.000000	1.000000	139.000000	10.000000	1.0	1.000000	20.000000
75%	20659.500000	14.000000	1.000000	207.000000	15.000000	1.0	1.000000	20.000000
max	27546.000000	40.000000	1.000000	276.000000	40.000000	1.0	1.000000	40.000000

Obtención de número de estaciones con "total" de 30

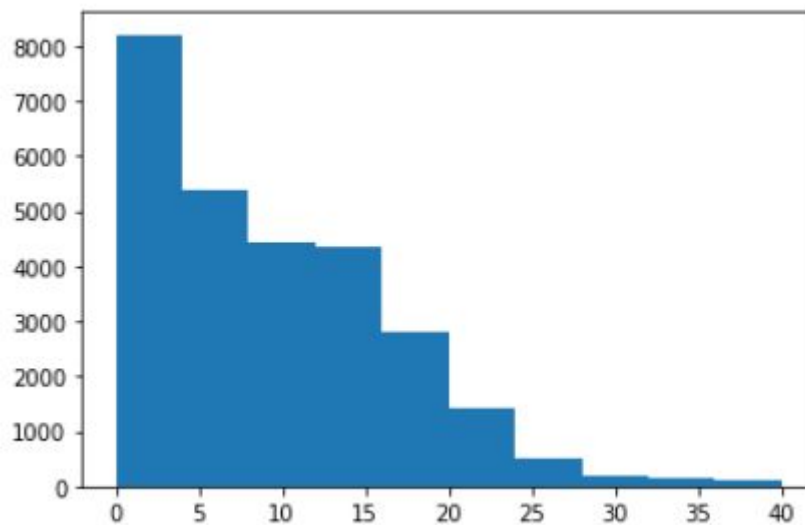
```
Out[7]: 1098
```

Número de estación con la media más alta de bicis disponibles

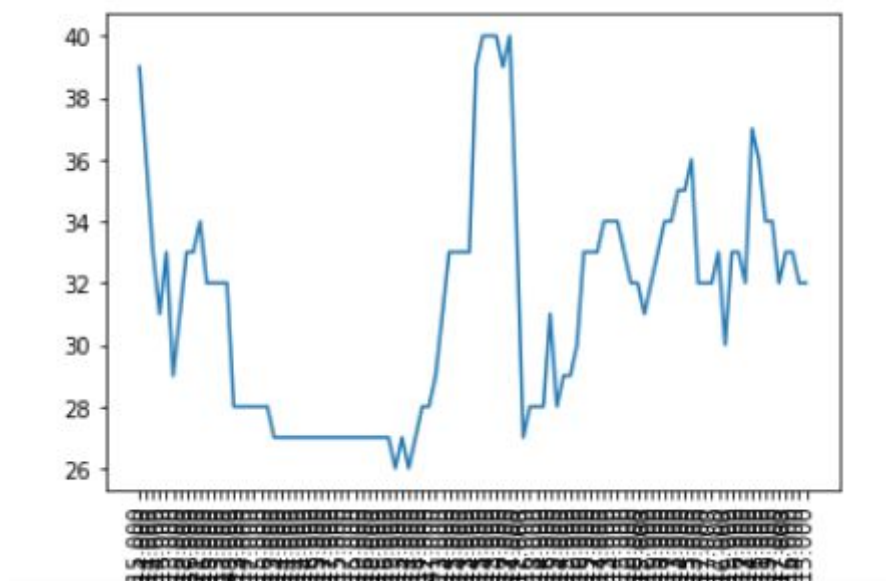
estation	average_available
50	31.26

```
Out[36]: 50
```

## Histograma de la estación de bicis disponibles



## Gráfica con la línea temporal de bicis disponibles





## 5. Conclusiones

En esta actividad hemos visto cómo usar las librerías Numpy, Pandas y Matplotlib de Python sobre Anaconda Jupyter; hemos creado un DataFrame a partir de un fichero .csv para luego analizar sus datos y mostrar información en forma tanto textual como gráficamente.

El notebook, el csv de origen y el presente documento se encuentran publicados en GitHub:

[https://github.com/adrimarmol/06MIOT\\_adrimarmol.git](https://github.com/adrimarmol/06MIOT_adrimarmol.git)