Python para el análisis de datos- Lectura 8

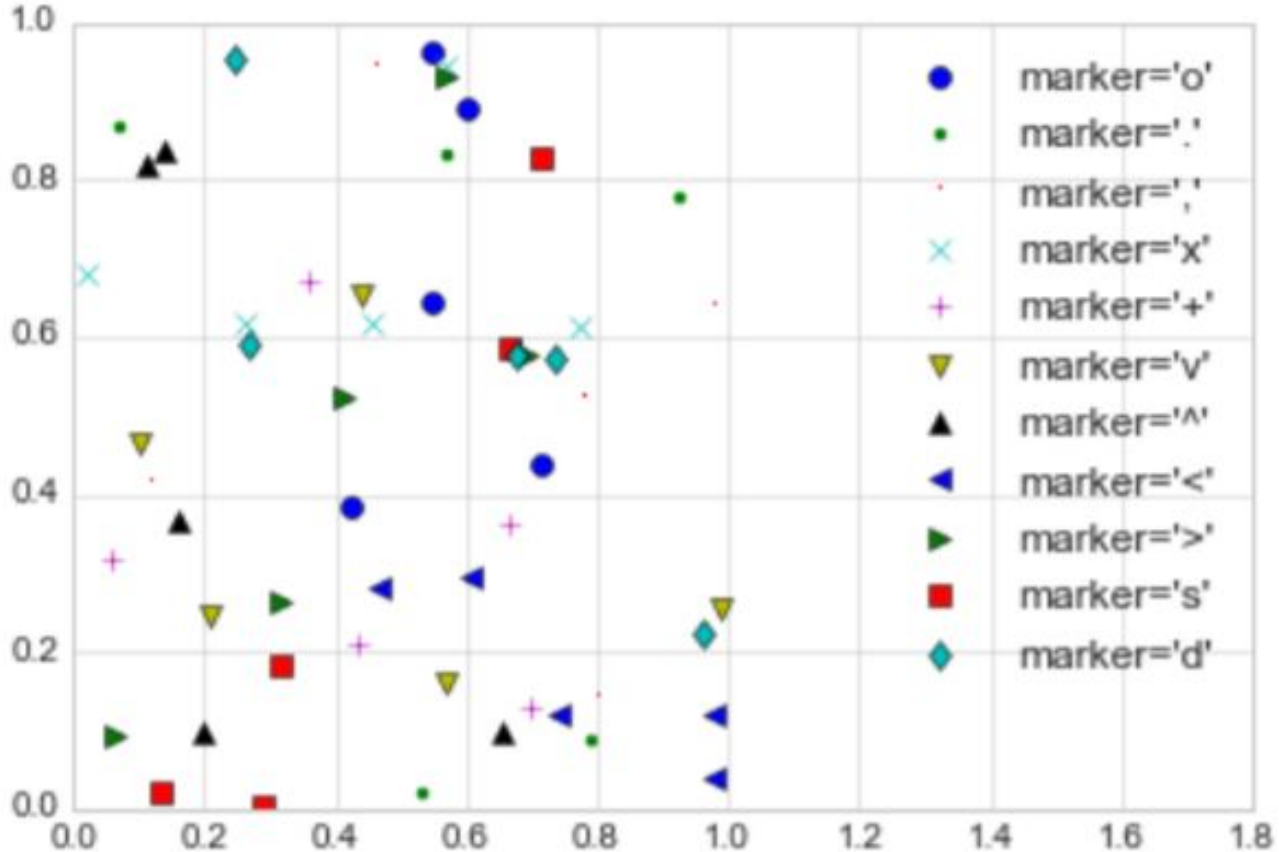
Ing. Pedro Rotta

Universidad de Piura - Vida Universitaria

Enero-2022

Gráficos de dispersión

Los gráficos de dispersión son un tipo de gráfico que no enlaza mediante segmentos los puntos de una traza, sino que señala los puntos representados individualmente con un punto, un círculo u otra forma.



Gráficos de dispersión

Existen varias formas de hacer un gráfico de dispersión. En primer lugar, mediante el método plot :

```
plt.plot(x,y,'o',color = 'black')
```

Donde el tercer valor, define el tipo de marcador y "color" define el color que se colocará.

Otra forma de definir un gráfico de dispersión es mediante el método scatter:

```
plt.scatter(x,y,marker = 'o')
```

Ver problema 2.

Histogramas

Otra forma de representar datos gráficamente es mediante histogramas.

El método para representar un histograma es hist. Cuya sintaxis se describe de la forma:

```
plt.hist(data,bins = n, normed = True, alpha = m, color = 'color')
```

Ver problema 3.

Texto y anotaciones

También dentro de la visualización de datos, es importante tener texto para señalar o marcar algo importante. La sintaxis para estos casos dados, se presenta en 2 partes. Primero el estilo:

```
style = dict(size=10, color='gray')
```

Y luego el texto, que queda definido como:

```
ax.text(xpoint, ypoint, Text, **style)
```

Ver problema 4.

Texto y anotaciones

También dentro de la visualización de datos, es importante tener texto para señalar o marcar algo importante. La sintaxis para estos casos dados, se presenta en 2 partes. Primero el estilo:

```
style = dict(size=10, color='gray')
```

Y luego el texto, que queda definido como:

```
ax.text(xpoint, ypoint, Text, **style)
```

Ver problema 4.

También se puede colocar texto y guiarlo mediante una flecha con la función annotate:

```
ax.anotate(text, xy = (), xytext= (),
arrowprops = dict(arrowstyle = "->",facecolor =
'black'))
```

Ver problema 5.

Seaborn

Seaborn es una librería basada en Matplotlib que permite hacer gráficas estadísticas especializadas de manera rápida y sencilla.

Como está construida sobre la base de Matplotlib, será importante al momento de importar seaborn, importar también matplotlib.

```
import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns
```

Seaborn ofrece como modelos de datasets o dataframes, para analizar en su método $\text{load}_d ataset$.

```
sns.load_dataset('tips')
```

Podemos ver los tipos de datasets en
`https://github.com/mwaskom/seaborn-data`

Gráficos de dispersión

Dado un dataframe, podemos usar el comando relplot para visualizar un gráfico de dispersión entre 2 variables, por ejemplo para el caso del dataframe tips:

```
sns.relplot(x = 'total_bill',y = 'tip',data = tips)
```

Para obtener un diferencial según si fuma o no fuma

```
sns.relplot(x = 'total_bill', hue = 'smoker',y =
'tip',data = tips)
```

Otra forma de sintaxis es:

```
sns.relplot(x = 'total_bill',y = 'tip',style =
'smoker', data = tips)
```

O para el tamaño:

```
sns.relplot(x = 'total_bill',y = 'tip',size =
'size', data = tips)
```

Ver problema 6.

Verifiación de datos

Algo que es importante de ver cuando se usa seaborn es verificar algunos datos estadísticos. Por ejemplo, llamemos a un dataframe de seaborn.

```
sns.load_dataset('fmri')
```

Podemos usar el comando value_counts() de pandas para obtener el total de valores por timepoint.

```
fmri['timepoint'].value_counts()
```

Un query permite filtar datos de un dataset de manera precisa, por ejemplo, dado este ejemplo, tenemos:

```
fmri.query('timepoint==0')['signal'].describe()
```

Ver problema 7.

Diagramas de distribución

Seaborn también ofrece la forma de definir diagramas de distribución como sigue a continuación:

```
sns.displot(dataset, x = column)
```

O también con diferenciador:

```
sns.displot(dataset, x = column, hue=column2)
```