

### Introducción

`pandas.plotting` es un submódulo en Pandas para gráficos avanzados, como gráficos de dispersión y matrices de correlación. Se usa para análisis exploratorio de datos.

**Bootstrap** es una técnica de remuestreo que estima la distribución de una estadística sin suponer una distribución específica. Es ideal para calcular intervalos de confianza cuando los datos son limitados.

### Funciones Clave

Función	Descripción
<code>scatter_matrix()</code>	Crea una matriz de dispersión para comparar todas las columnas numéricas en un DataFrame.
<code>andrews_curves()</code>	Visualiza patrones en un DataFrame usando curvas de Andrews.
<code>parallel_coordinates()</code>	Muestra las columnas de un DataFrame como líneas paralelas para ver agrupaciones.

### Funciones Clave (cont)

`lag_plot()` Grafica una variable contra sus valores desplazados para ver autocorrelación.

### ¿Cuándo usar Bootstrap?

- **Datos limitados:** Si tienes pocos datos, Bootstrap permite estimar intervalos de confianza.

- **Estimación de intervalos de confianza:** Útil cuando se desconoce la distribución de los datos.

- **Sin supuestos paramétricos:** No necesitas asumir normalidad u otras propiedades de los datos.

### Código Básico: Bootstrap con pandas.plotting

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from pandas.plotting import scatter_matrix

# Datos de ejemplo
data = pd.DataFrame({
    'X': np.random.randn(50, 10),
    'Y': np.random.randn(50, 10),
})

# Aplicar Bootstrap
bootstrap_means = []
samples = 1000 # Número de muestras
for _ in range(samples):
```

### Código Básico: Bootstrap con pandas.plotting (cont)

```
> sample = data.sample(frac=1, replace=True)
bootstrap_means.append(sample.mean())
bootstrap_df = pd.DataFrame(bootstrap_means)

# Visualización
scatter_matrix(bootstrap_df, alpha=0.5, diagonal='kde')
plt.show()
```

### Personalización de Gráficos en Matplotlib

Elemento	Código
Título Central	<code>plt.suptitle("Título", fontsize=16, fontweight='bold', color='navy')</code>
Título Ejes X/Y	<code>plt.xlabel("Eje X"), plt.ylabel("Eje Y")</code>
Cambiar Colores	<code>color='teal'</code> (en funciones de Matplotlib y <code>pandas.plotting</code> )
Leyendas	<code>plt.legend(["Label 1", "Label 2"])</code> para etiquetas automáticas, o <code>ax.text</code> para leyendas personalizadas.

### Ejemplo Completo

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from pandas.plotting import scatter_matrix

# Generar datos
np.random.seed(0)
data = pd.DataFrame({
    'A': np.random.randn(60, 15, 200),
    'B': np.random.randn(50, 10, 200)
})

# Aplicar Bootstrap
bootstrap_means = []
samples = 1000 # Número de muestras Bootstrap
for _ in range(samples):
    sample = data.sample(frac=1, replace=True)
    bootstrap_means.append(sample.mean())

bootstrap_df = pd.DataFrame(bootstrap_means)

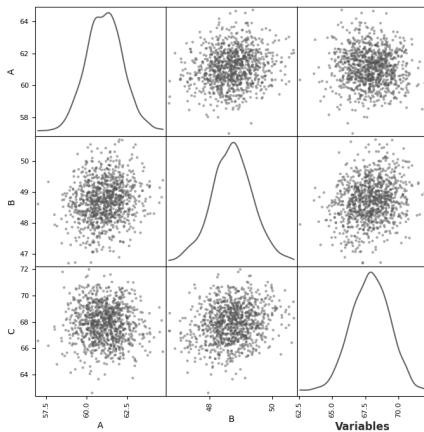
# Visualización y personalización
fig, ax = plt.subplots(figsize=(10, 8))
scatter_matrix(bootstrap_df, alpha=0.5, figsize=(8, 8), diagonal='kde', color='teal')
```

### Ejemplo Completo (cont)

```
> plt.suptitle("Distribución
Bootstrap de Medias", fontsi-
ze=16, fontweight='bold',
color='navy')
plt.xlabel("Variables", fontsi-
ze=12, fontweight='bold',
color='darkred')
plt.ylabel("Valores de la Media",
fontsi=12, fontweight='bold',
color='darkred')
# Leyenda personalizada
for i, col in enumerate(bootstra-
p_df.columns):
    ax.text(0.9, 0.1 - i*0.1, f"Media
de {col}", transform=ax.trans-
Axes, fontsize=12, color='teal')
plt.show()
```

### imagenes

Distribución Bootstrap de Medias para A, B y C



C

By **Comtive45**  
[cheatography.com/comtive45/](https://cheatography.com/comtive45/)

Not published yet.  
 Last updated 3rd November, 2024.  
 Page 2 of 2.

Sponsored by **CrosswordCheats.com**  
 Learn to solve cryptic crosswords!  
<http://crosswordcheats.com>