Covariance

```
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
plt.style.use('ggplot')

df = pd.DataFrame()

from re import X
x = pd.Series([20,45,62,88,105])
y = pd.Series([35,60,78,99,115])

df['x'] = x
df['y'] = y

df
```

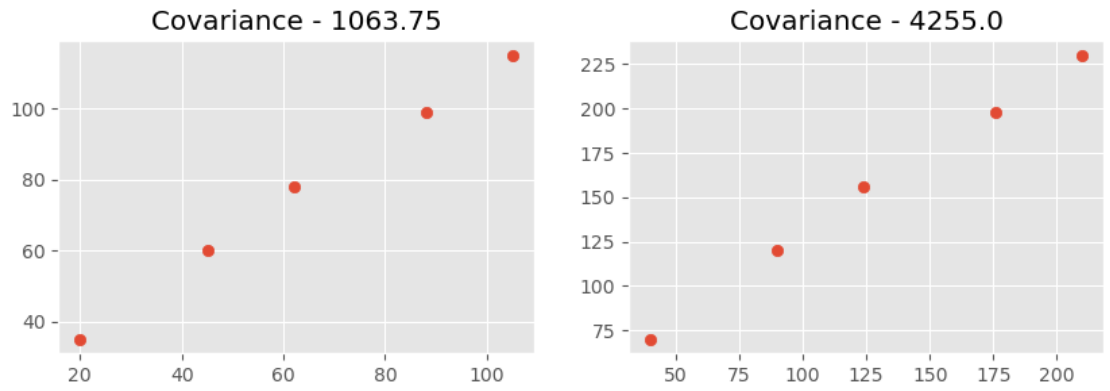
```
     x    y
0   20   35
1   45   60
2   62   78
3   88   99
4  105  115
```

```
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(10, 3))

# Plot scatterplots on each axes
ax1.scatter(df['x'], df['y'])
ax2.scatter(df['x']*2, df['y']*2)

ax1.set_title("Covariance - " + str(np.cov(df['x'],df['y'])[0,1]))
ax2.set_title("Covariance - " + str(np.cov(df['x']*2,df['y']*2)[0,1]))

Text(0.5, 1.0, 'Covariance - 4255.0')
```



Explanation of above code

The code in the FOCAL CELL is responsible for creating a figure with two subplots (ax1 and ax2) using the plt.subplots function. The figure has a size of 10 units wide and 3 units tall.

Next, scatter plots are created on each of the axes. In ax1, the scatter plot is created using the scatter function, where the x-coordinates are taken from the 'x' column of the DataFrame df, and the y-coordinates are taken from the 'y' column of df. Similarly, in ax2, the scatter plot is created with the x-coordinates multiplied by 2 and the y-coordinates multiplied by 2.

The set_title function is then used to set the titles of both axes. In ax1, the title is set as "Covariance - " concatenated with the value of the covariance between 'x' and 'y' calculated using np.cov(df['x'], df['y'])[0,1]. Similarly, in ax2, the title is set as "Covariance - " concatenated with the value of the covariance between 'x' (multiplied by 2) and 'y' (multiplied by 2) calculated using np.cov(df['x']*2, df['y']*2)[0,1].

The resulting figure is not displayed or printed directly. If the code were to be executed, it would generate a figure with two scatter plots and titles indicating the covariance values for each plot. The result of the FOCAL CELL is a text output showing the title of ax2, which is "Covariance - 4255.0".

```
print(np.cov(df['x'],df['y'])[0,1])
print(np.cov(df['x']*2,df['y']*2)[0,1])
```

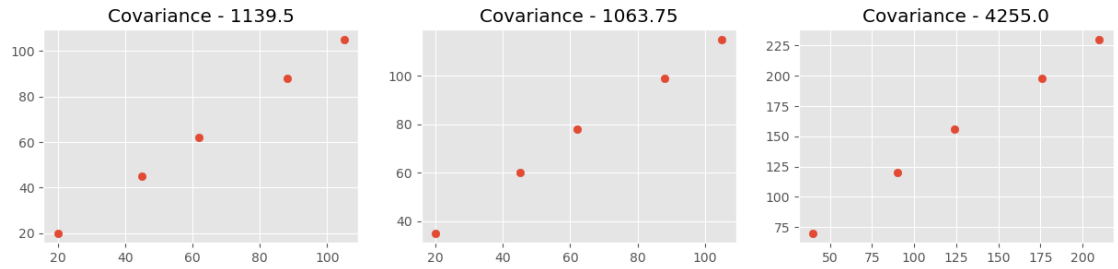
```
1063.75
4255.0
```

```
fig, ax = plt.subplots(1, 3, figsize=(15, 3))

# Plot scatterplots on each axes
ax[0].scatter(df['x'], df['x'])
ax[1].scatter(df['x'], df['y'])
ax[2].scatter(df['x']*2, df['y']*2)

ax[0].set_title("Covariance - " + str(np.cov(df['x'],df['x'])[0,1]))
ax[1].set_title("Covariance - " + str(np.cov(df['x'],df['y'])[0,1]))
ax[2].set_title("Covariance - " + str(np.cov(df['x']*2,df['y']*2)
[0,1]))
```

```
Text(0.5, 1.0, 'Covariance - 4255.0')
```



Explanation of above code

The code in the FOCAL CELL is generating a figure with three scatterplots and calculating the covariance for each plot. Let's break it down line by line:

fig, ax = plt.subplots(1, 3, figsize=(15, 3))

This line creates a figure with 1 row and 3 columns of subplots, and sets the figure size to 15 units wide and 3 units tall.

ax[0].scatter(df['x'], df['x']) ax[1].scatter(df['x'], df['y']) ax[2].scatter(df['x']*2, df['y']*2)

These lines plot scatterplots on each of the three subplots.

ax[0] plots the values from the 'x' column against itself. ax[1] plots the values from the 'x' column against the values from the 'y' column. ax[2] plots the values from the 'x' column multiplied by 2 against the values from the 'y' column multiplied by 2. ax[0].set_title("Covariance - " + str(np.cov(df['x'],df['x'])[0,1])) ax[1].set_title("Covariance - " + str(np.cov(df['x'],df['y'])[0,1])) ax[2].set_title("Covariance - " + str(np.cov(df['x']*2,df['y']*2)[0,1]))

These lines set the titles for each subplot. The title includes the string "Covariance - " concatenated with the covariance value calculated using np.cov() function.

ax[0] displays the covariance between the 'x' column and itself. ax[1] displays the covariance between the 'x' column and the 'y' column. ax[2] displays the covariance between the 'x' column multiplied by 2 and the 'y' column multiplied by 2. The output displayed in the STDOUT for this code is empty, as there are no print statements in the FOCAL CELL. However, the resulting figure shows the scatterplots and the titles of each subplot, with the third subplot displaying "Covariance - 4255.0" as its title.

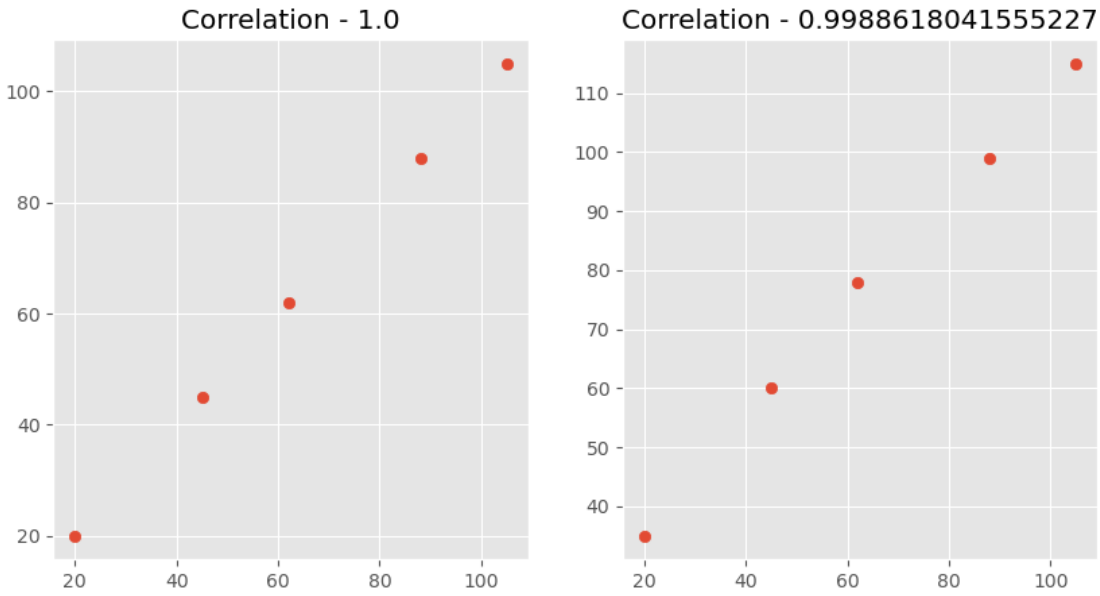
Correlation

```
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(10, 5))

# Plot scatterplots on each axes
ax1.scatter(df['x'], df['x'])
ax2.scatter(df['x'], df['y'])

ax1.set_title("Correlation - " + str(df['x'].corr(df['x'])))
ax2.set_title("Correlation - " + str((df['x']).corr(df['y'])))
```

```
Text(0.5, 1.0, 'Correlation - 0.9988618041555227')
```



Explanation of above code

Importing necessary libraries:

The code begins by importing the required libraries, including seaborn, pandas, matplotlib.pyplot, and numpy. These libraries are commonly used for data manipulation and visualization in Python. Creating a DataFrame:

The code creates an empty DataFrame called df using the pd.DataFrame() function from the pandas library. This DataFrame will be used to store the data. Defining the data:

Two pd.Series objects, x and y, are created and assigned with a series of numerical values. These series represent the values of two variables, x and y, respectively. Adding data to the DataFrame:

The code adds the x and y series as columns to the df DataFrame using df['x'] = x and df['y'] = y. Creating subplots for scatterplots:

The code creates a figure with two subplots using the plt.subplots() function, specifying a 1x2 grid of subplots and a figure size of 10x5. The subplots are assigned to ax1 and ax2 variables. Plotting scatterplots:

The code plots scatterplots on each of the subplots using the scatter() function. ax1.scatter(df['x'], df['x']) plots the values of x against itself, creating a diagonal line. ax2.scatter(df['x'], df['y']) plots the values of x against y, creating a scatterplot. Setting titles for subplots:

The code sets the titles for the subplots using the set_title() function. ax1.set_title("Correlation - " + str(df['x'].corr(df['x']))) sets the title for ax1 subplot with the correlation coefficient between x and itself. ax2.set_title("Correlation - " +

str((df['x']).corr(df['y']))) sets the title for ax2 subplot with the correlation coefficient between x and y. Output:

The resulting scatterplots with their respective correlation coefficients are displayed. The str(df['x'].corr(df['x'])) and str((df['x']).corr(df['y'])) are used to convert the calculated correlation coefficients to strings for concatenation with the title strings.

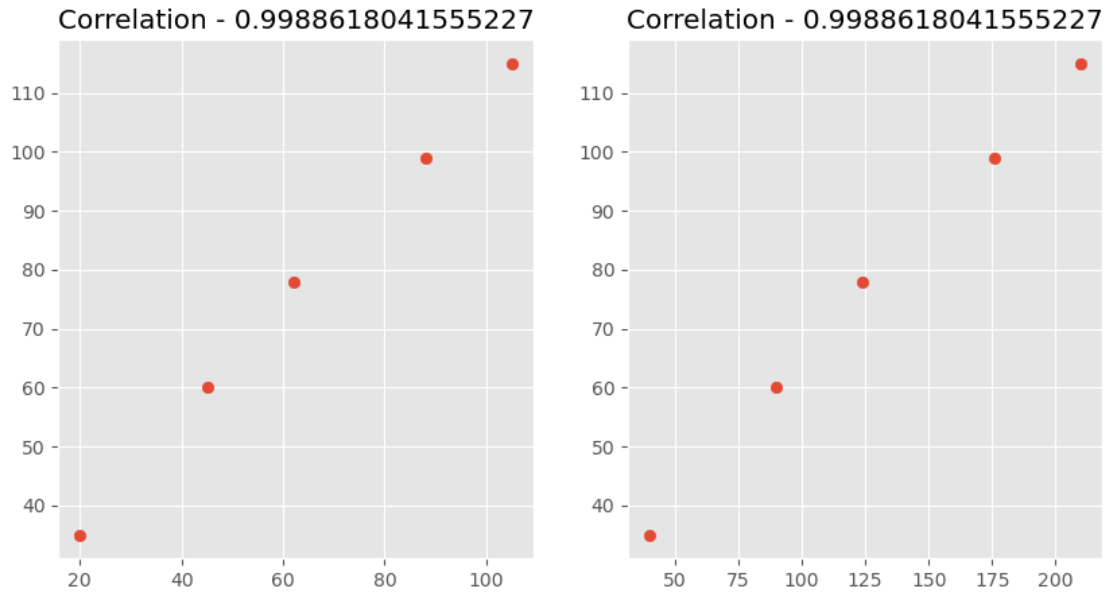
The output of the FOCAL CELL is a visualization with two subplots, each displaying a scatterplot and its corresponding correlation coefficient in the title. The correlation coefficient represents the strength and direction of the linear relationship between the variables.

```
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(10, 5))

# Plot scatterplots on each axes
ax1.scatter(df['x'], df['y'])
ax2.scatter(df['x']*2, df['y'])

ax1.set_title("Correlation - " + str(df['x'].corr(df['y'])))
ax2.set_title("Correlation - " + str((df['x']*2).corr(df['y']*2)))
```

```
Text(0.5, 1.0, 'Correlation - 0.9988618041555227')
```



Explanation of above code

The code in the focal cell is responsible for creating a figure with two subplots (side-by-side) using the plt.subplots() function from the matplotlib library. Each subplot represents a scatter plot.

The first subplot is referenced as ax1, and the second subplot is referenced as ax2. The size of the figure is set to 10 units wide and 5 units tall using the figsize parameter.

The code then plots a scatter plot on each of the subplots using the scatter() function. The scatter plots are created by passing the 'x' and 'y' columns of the 'df' DataFrame as the x and y coordinates, respectively. In the first subplot, the 'x' and 'y' columns are used directly, while in the second subplot, the 'x' column is multiplied by 2 before plotting.

Next, the code sets the titles for each subplot using the set_title() function. The titles include the string "Correlation - " followed by the value of the correlation coefficient calculated using the corr() function. In the first subplot, the correlation coefficient is calculated between the 'x' and 'y' columns of the DataFrame. In the second subplot, the correlation coefficient is calculated between the 'x' column multiplied by 2 and the 'y' column.

Finally, the resulting figure with the scatter plots and titles is displayed.

The output of the focal cell, which is not shown in the STDOUT, is the text "Correlation - 0.9988618041555227", indicating the correlation coefficient between the 'x' and 'y' columns of the DataFrame.