

```
In [1]: import pandas as pd  
import matplotlib.pyplot as plt
```

## Documentation

- Import the necessary library: import matplotlib.pyplot as plt.
- Create a figure and a set of subplots with a size of 12 inches by 4 inches: fig, axs = plt.subplots(figsize=(12, 4)).
- Plot the 'air' data as an area plot on the specified axes: air.plot.area(ax=axs).
- Add labels to the X and Y axes, and set the title for the plot.
  - Display the plot using plt.show().

```
In [2]: air= pd.read_csv("air (Matplotlib).csv")
```

```
In [3]: air.head()
```

```
Out[3]:
```

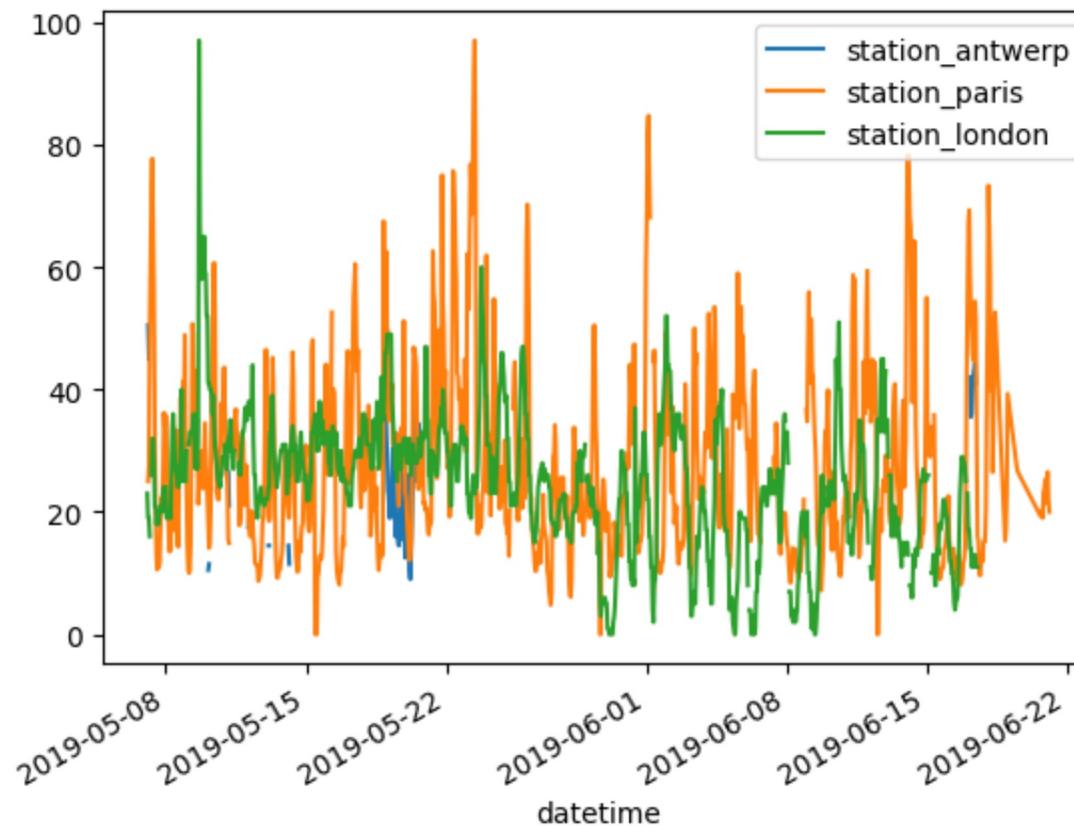
	datetime	station_antwerp	station_paris	station_london
0	2019-05-07 02:00:00	NaN	NaN	23.0
1	2019-05-07 03:00:00	50.5	25.0	19.0
2	2019-05-07 04:00:00	45.0	27.7	19.0
3	2019-05-07 05:00:00	NaN	50.4	16.0
4	2019-05-07 06:00:00	NaN	61.9	NaN

```
In [4]: air= pd.read_csv("air (Matplotlib).csv", index_col=0, parse_dates=True)  
air.head()
```

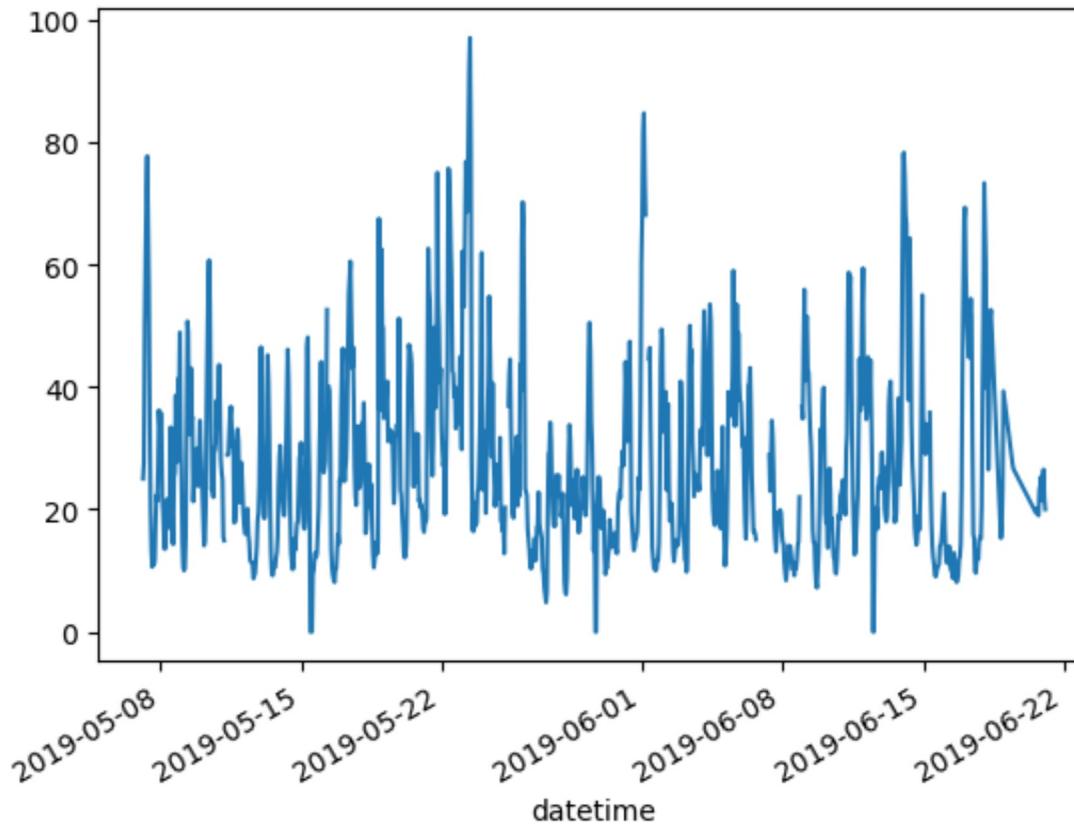
Out[4]:

	station_antwerp	station_paris	station_london
datetime			
2019-05-07 02:00:00	NaN	NaN	23.0
2019-05-07 03:00:00	50.5	25.0	19.0
2019-05-07 04:00:00	45.0	27.7	19.0
2019-05-07 05:00:00	NaN	50.4	16.0
2019-05-07 06:00:00	NaN	61.9	NaN

```
In [5]: air.plot()  
plt.show()
```

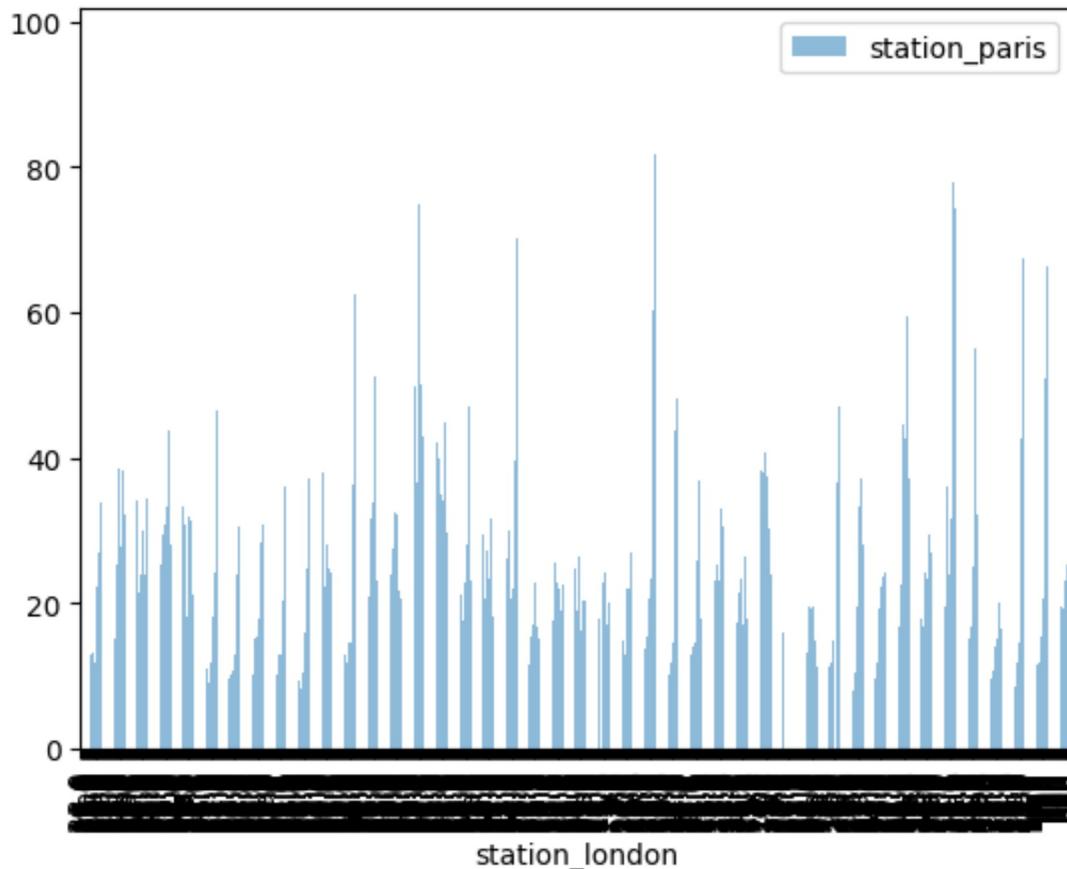


```
In [6]: air["station_paris"].plot()  
Out[6]: <Axes: xlabel='datetime'>
```



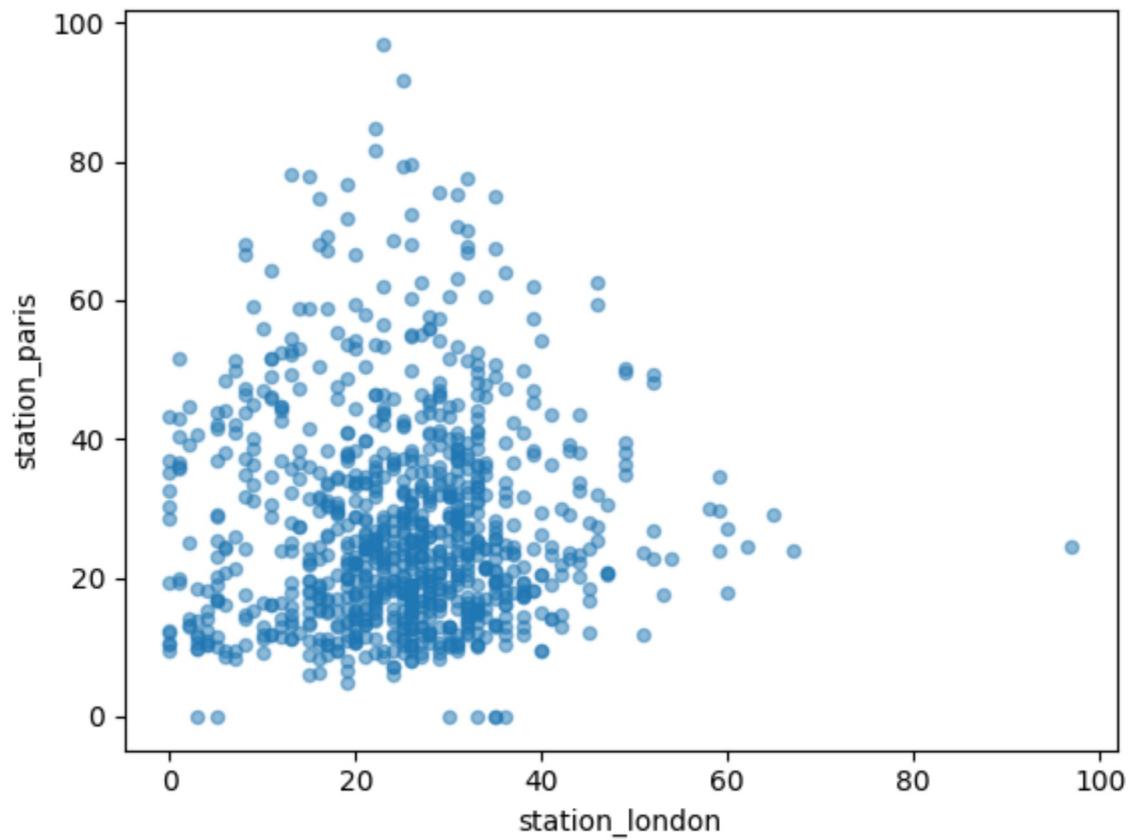
```
In [7]: air.plot.bar(x="station_london", y="station_paris", alpha=0.5)
```

```
Out[7]: <Axes: xlabel='station_london'>
```



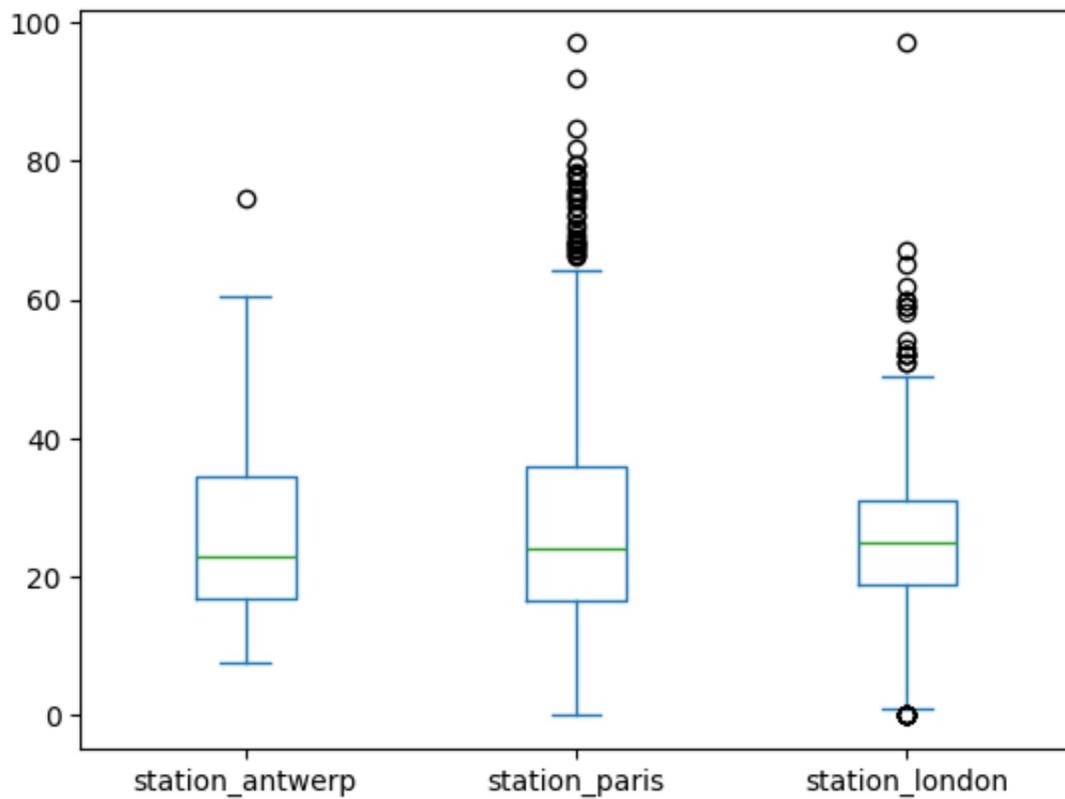
```
In [8]: air.plot.scatter(x="station_london", y="station_paris", alpha=0.5)
```

```
Out[8]: <Axes: xlabel='station_london', ylabel='station_paris'>
```

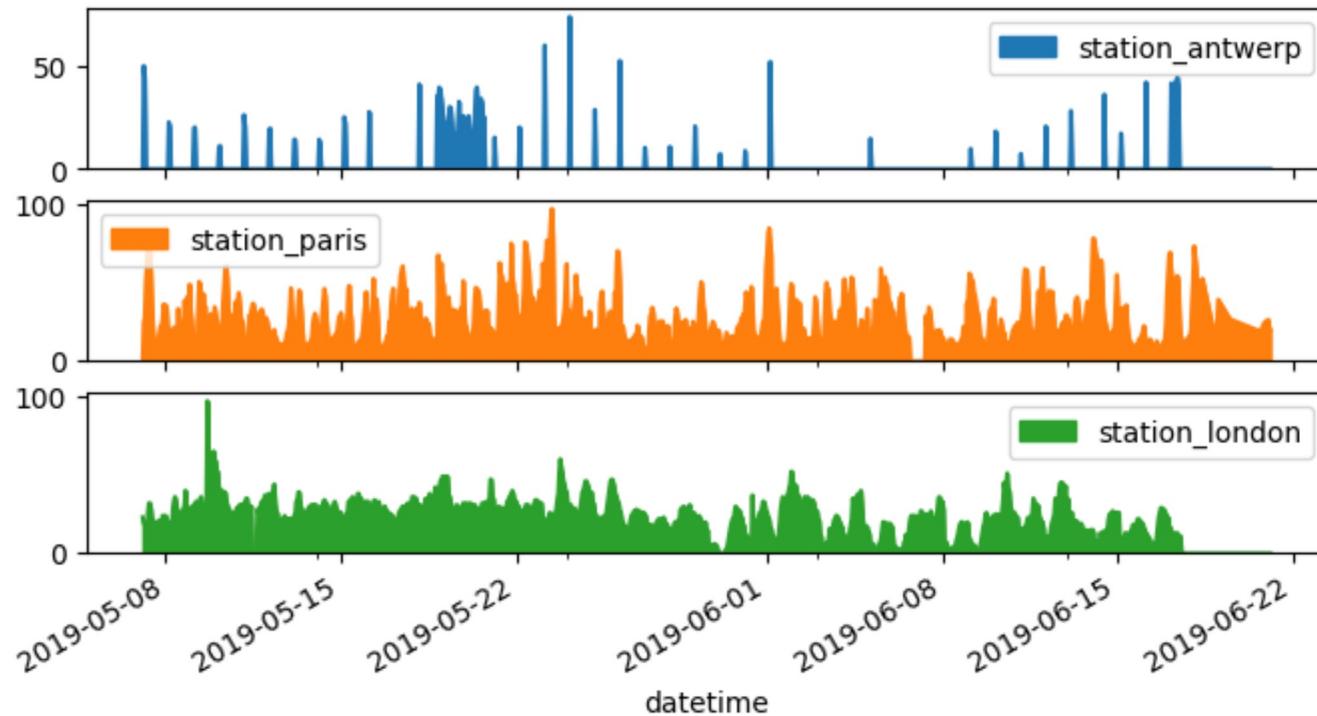


```
In [9]: air.plot.box()
```

```
Out[9]: <Axes: >
```



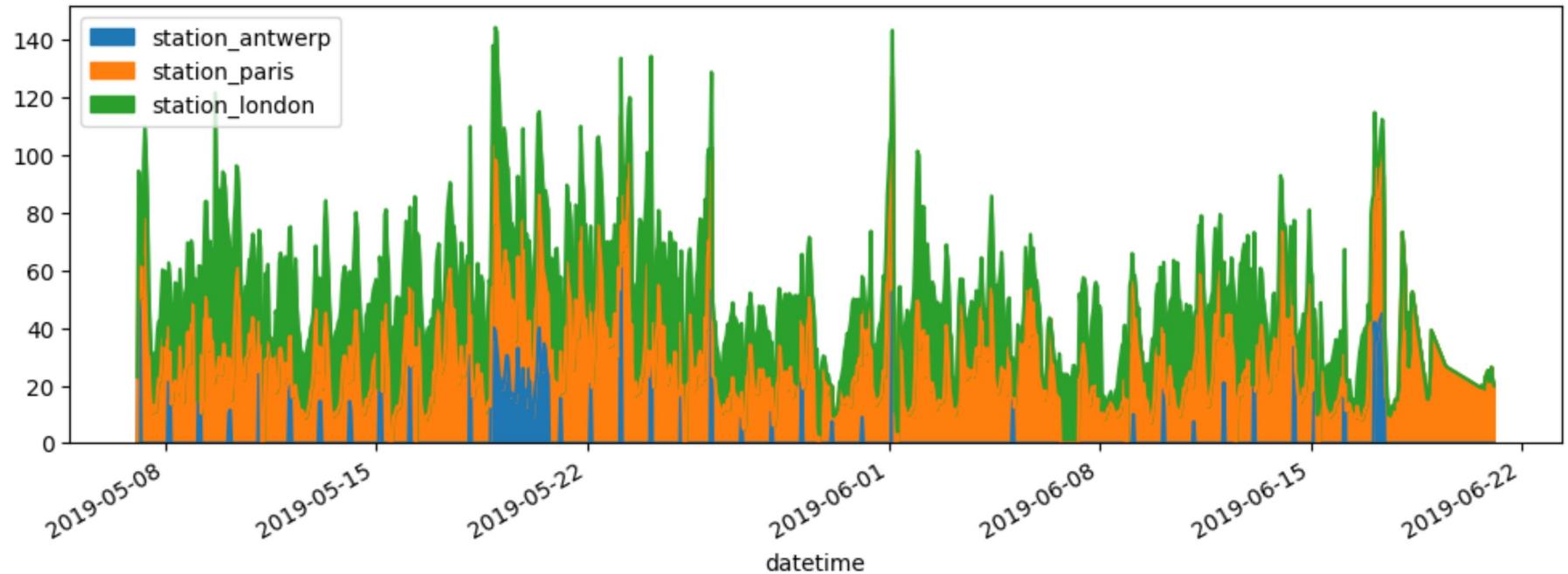
```
In [10]: axs = air.plot.area(figsize=(8,4), subplots=True)
```



```
In [11]: fig, axs = plt.subplots(figsize=(12, 4))

air.plot.area(ax=axs)

Out[11]: <Axes: xlabel='datetime'>
```



```
In [13]: air.head()
```

```
Out[13]:
```

datetime	station_antwerp	station_paris	station_london
2019-05-07 02:00:00	NaN	NaN	23.0
2019-05-07 03:00:00	50.5	25.0	19.0
2019-05-07 04:00:00	45.0	27.7	19.0
2019-05-07 05:00:00	NaN	50.4	16.0
2019-05-07 06:00:00	NaN	61.9	NaN

Create a new column:

```
In [19]: air["New column"] = air["station_london"]*1.881
```

```
In [20]: air.head()
```

```
Out[20]:
```

	station_antwerp	station_paris	station_london	New column
datetime				
2019-05-07 02:00:00	NaN	NaN	23.0	43.263
2019-05-07 03:00:00	50.5	25.0	19.0	35.739
2019-05-07 04:00:00	45.0	27.7	19.0	35.739
2019-05-07 05:00:00	NaN	50.4	16.0	30.096
2019-05-07 06:00:00	NaN	61.9	NaN	NaN

## Rename the columns:

```
In [27]: air_renamed = air.rename(  
    columns={  
        "station_antwerp": "Column1",  
        "station_paris": "Column2",  
        "station_london": "Column3"  
    }  
)
```

- The rename method is used to rename columns in the DataFrame air.
- The columns parameter is a dictionary where keys are the current column names, and values are the new column names.
- Column names "station\_antwerp," "station\_paris," and "station\_london" are renamed to "Column1," "Column2," and "Column3," respectively.
- The resulting DataFrame with the new column names is stored in the variable "air\_renamed"

```
In [29]: air_renamed.head()
```

Out[29]:

	Column1	Column2	Column3	New column
datetime				
2019-05-07 02:00:00	NaN	NaN	23.0	43.263
2019-05-07 03:00:00	50.5	25.0	19.0	35.739
2019-05-07 04:00:00	45.0	27.7	19.0	35.739
2019-05-07 05:00:00	NaN	50.4	16.0	30.096
2019-05-07 06:00:00	NaN	61.9	NaN	NaN

## Add new column divide c3 and new column

In [35]: `air_renamed["average_column"] = (air_renamed["Column3"] / air_renamed["New column"])`

- Create a new column named "average\_column" in the DataFrame
- The values in this column are calculated by dividing the values in "Column3" by the values in "New column"

In [41]: `air_renamed.head()`

Out[41]:

	station_antwerp	station_paris	station_london	new column
datetime				
2019-05-07 02:00:00	NaN	NaN	23.0	43.263
2019-05-07 03:00:00	50.5	25.0	19.0	35.739
2019-05-07 04:00:00	45.0	27.7	19.0	35.739
2019-05-07 05:00:00	NaN	50.4	16.0	30.096
2019-05-07 06:00:00	NaN	61.9	NaN	NaN

- Convert the column names of air\_renamed to lowercase and create a new DataFrame named air\_renamed2

```
In [48]: # Creating a new DataFrame 'air_renamed2' with Lowercase column names
air_renamed2 = air_renamed.rename(columns=str.lower)

# Displaying the first few rows of the 'air_renamed2' DataFrame
air_renamed2.head()
```

Out[48]:

	column1	column2	column3	new column	average_column
datetime					
2019-05-07 02:00:00	NaN	NaN	23.0	43.263	0.531632
2019-05-07 03:00:00	50.5	25.0	19.0	35.739	0.531632
2019-05-07 04:00:00	45.0	27.7	19.0	35.739	0.531632
2019-05-07 05:00:00	NaN	50.4	16.0	30.096	0.531632
2019-05-07 06:00:00	NaN	61.9	NaN	NaN	NaN

In [ ]:

In [ ]: