Clean your time series data

VISUALIZING TIME SERIES DATA IN PYTHON

Thomas Vincent

Head of Data Science, Getty Images

datacamp

The CO2 level time series

A snippet of the weekly measurements of CO2 levels at the Mauna Loa Observatory, Hawaii.

```
datastamp    co2
1958-03-29  316.1
1958-04-05  317.3
1958-04-12  317.6
...
...
2001-12-15  371.2
2001-12-22  371.3
2001-12-29  371.5
```

Finding missing values in a DataFrame

```
print(df.isnull())
```

```
datestamp   co2
1958-03-29  False
1958-04-05  False
1958-04-12  False
```

```
print(df.notnull())
```

```
datestamp    co2
1958-03-29   True
1958-04-05   True
1958-04-12   True
...
```

Counting missing values in a DataFrame

```
print(df.isnull().sum())
```

```
datestamp     0
co2          59
dtype: int64
```

Replacing missing values in a DataFrame

```
print(df)
```

```
...
5  1958-05-03  316.9
6  1958-05-10    NaN
7  1958-05-17  317.5
...
```

```
df = df.fillna(method='bfill')
print(df)
```

```
...
5  1958-05-03  316.9
6  1958-05-10  317.5
7  1958-05-17  317.5
...
```

Let's practice!

VISUALIZING TIME SERIES DATA IN PYTHON

Plot aggregates of your data

VISUALIZING TIME SERIES DATA IN PYTHON

Thomas Vincent
Head of Data Science, Getty Images

Moving averages

- In the field of time series analysis, a moving average can be used for many different purposes:
  - smoothing out short-term fluctuations
  - removing outliers
  - highlighting long-term trends or cycles.

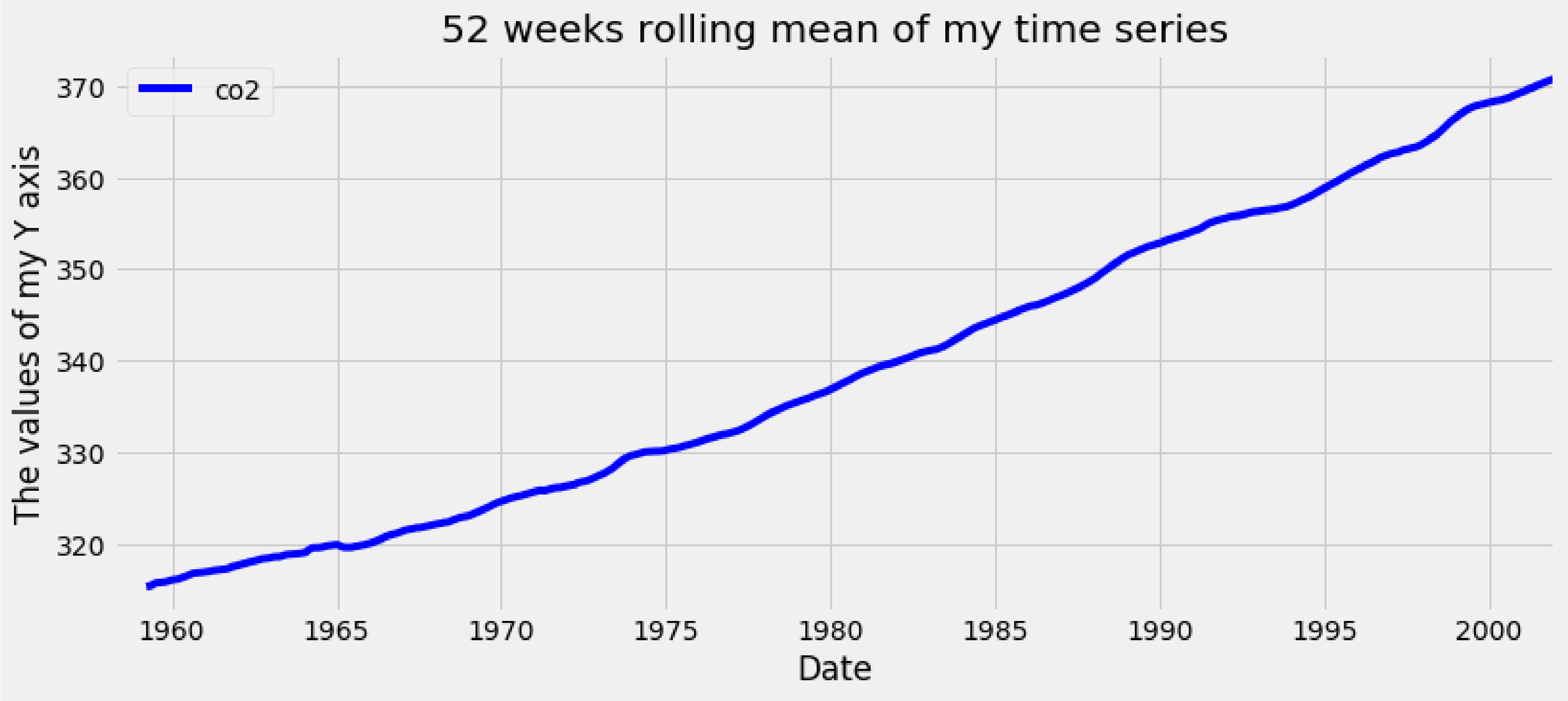
The moving average model

```
co2_levels_mean = co2_levels.rolling(window=52).mean()

ax = co2_levels_mean.plot()
ax.set_xlabel("Date")
ax.set_ylabel("The values of my Y axis")
ax.set_title("52 weeks rolling mean of my time series")

plt.show()
```

A plot of the moving average for the CO2 data



Computing aggregate values of your time series

```
co2_levels.index
```

```
DatetimeIndex(['1958-03-29', '1958-04-05',...],
              dtype='datetime64[ns]', name='datestamp',
              length=2284, freq=None)
```

```
print(co2_levels.index.month)
```

```
array([ 3,  4,  4, ..., 12, 12, 12], dtype=int32)
```

```
print(co2_levels.index.year)
```

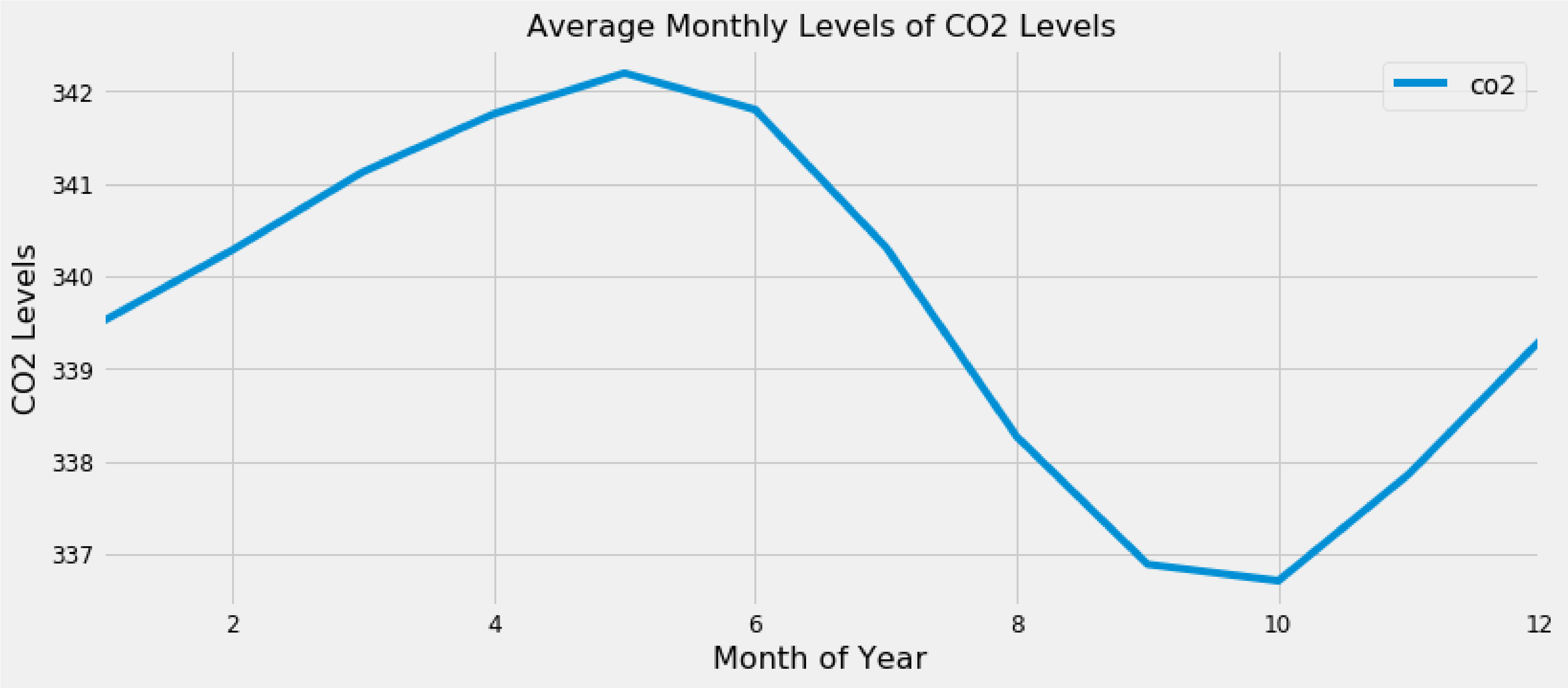
```
array([1958, 1958, 1958, ..., 2001,
      2001, 2001], dtype=int32)
```

Plotting aggregate values of your time series

```
index_month = co2_levels.index.month
co2_levels_by_month = co2_levels.groupby(index_month).mean()
co2_levels_by_month.plot()

plt.show()
```

Plotting aggregate values of your time series



Let's practice!

VISUALIZING TIME SERIES DATA IN PYTHON

Summarizing the values in your time series data

VISUALIZING TIME SERIES DATA IN PYTHON

Thomas Vincent
Head of Data Science, Getty Images

Obtaining numerical summaries of your data

- What is the average value of this data?
- What is the maximum value observed in this time series?

The `.describe()` method automatically computes key statistics of all numeric columns in your DataFrame

```
print(df.describe())
```

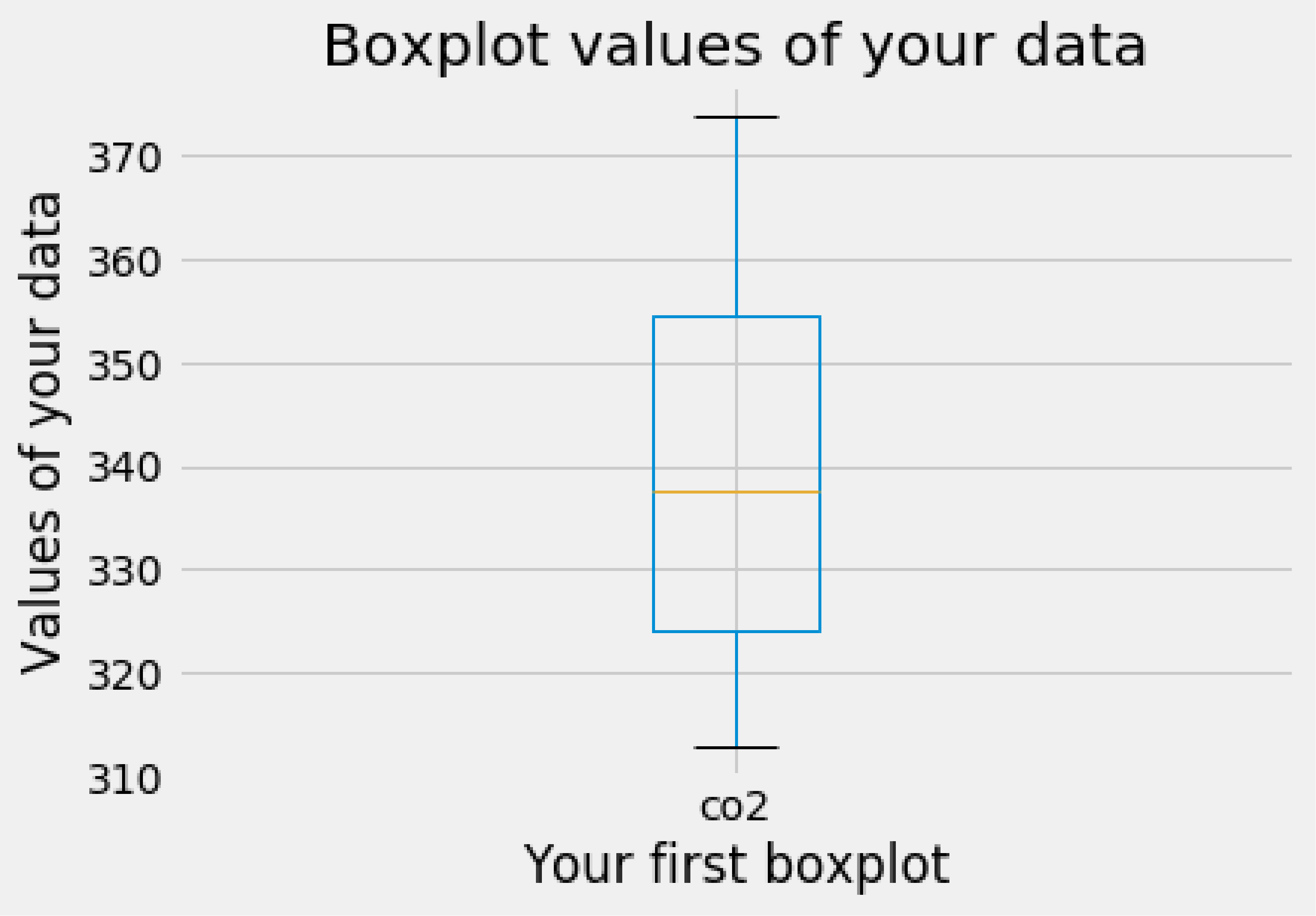
```
               co2
count  2284.000000
mean    339.657750
std      17.100899
min     313.000000
25%     323.975000
50%     337.700000
75%     354.500000
max     373.900000
```

Summarizing your data with boxplots

```
ax1 = df.boxplot()
ax1.set_xlabel('Your first boxplot')
ax1.set_ylabel('Values of your data')
ax1.set_title('Boxplot values of your data')

plt.show()
```

A boxplot of the values in the CO2 data

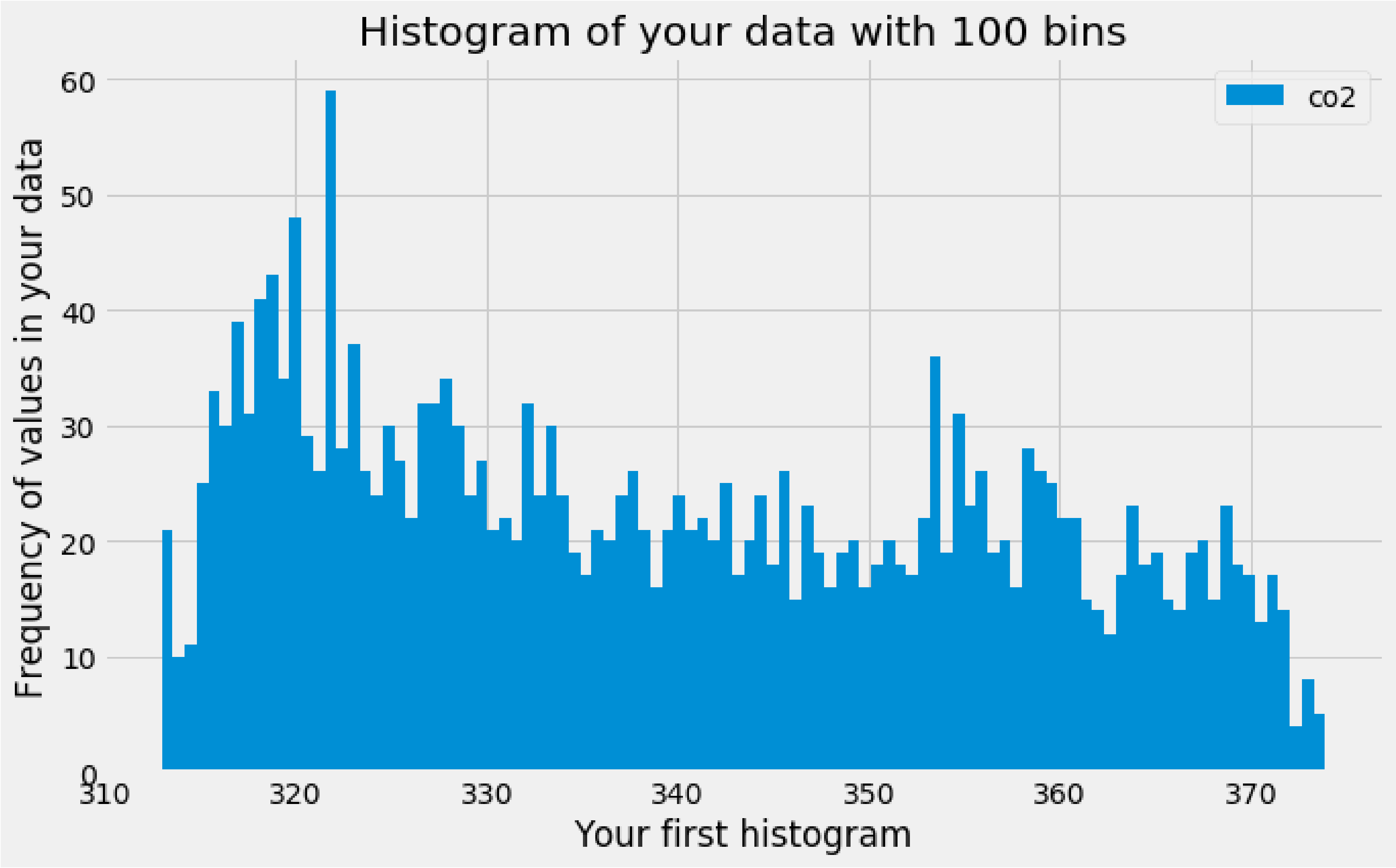


Summarizing your data with histograms

```
ax2 = df.plot(kind='hist', bins=100)
ax2.set_xlabel('Your first histogram')
ax2.set_ylabel('Frequency of values in your data')
ax2.set_title('Histogram of your data with 100 bins')

plt.show()
```

A histogram plot of the values in the CO2 data

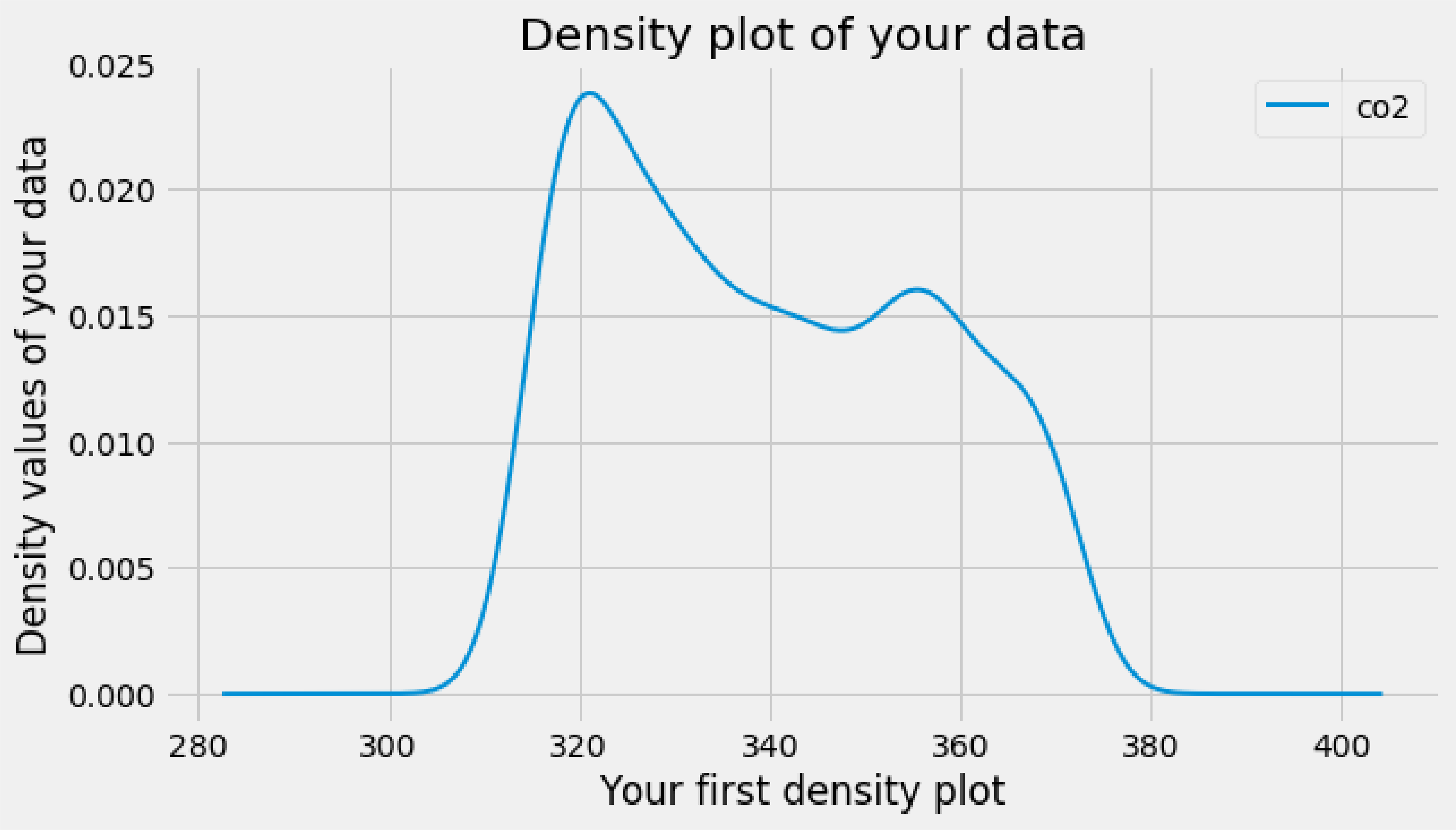


Summarizing your data with density plots

```
ax3 = df.plot(kind='density', linewidth=2)
ax3.set_xlabel('Your first density plot')
ax3.set_ylabel('Density values of your data')
ax3.set_title('Density plot of your data')

plt.show()
```

A density plot of the values in the CO2 data



Let's practice!

VISUALIZING TIME SERIES DATA IN PYTHON