Welcome to the course!

VISUALIZING TIME SERIES DATA IN PYTHON

Thomas Vincent
Head of Data Science, Getty Images

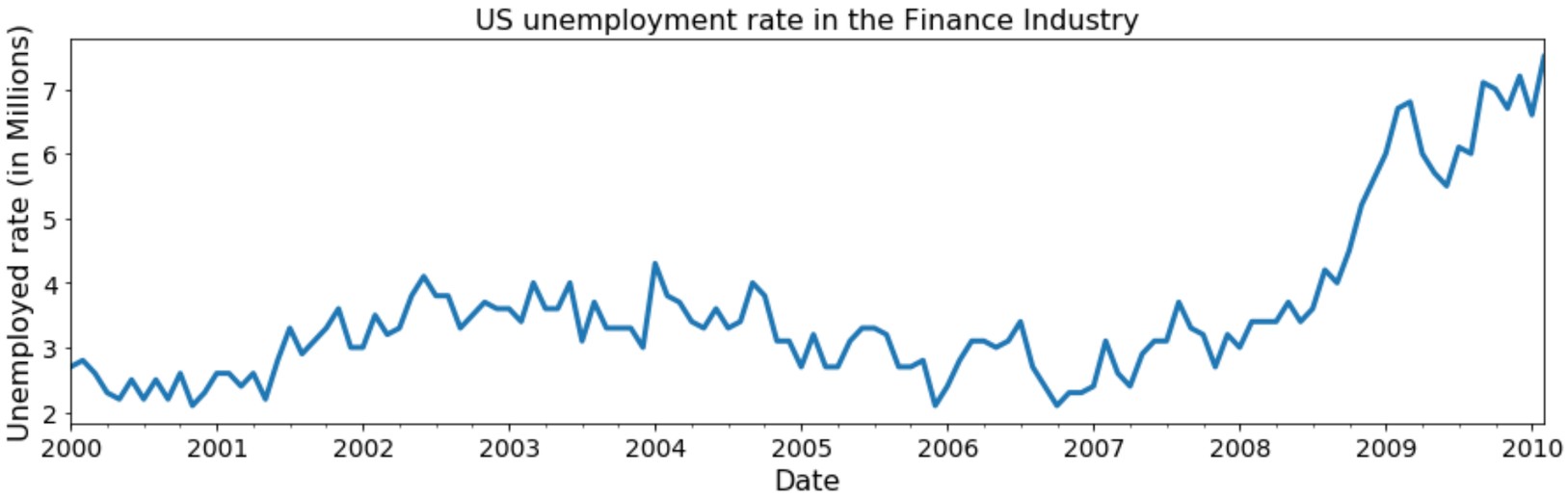
Prerequisites

- Intro to Python for Data Science
- Intermediate Python for Data Science

Time series in the field of Data Science

- Time series are a fundamental way to store and analyze many types of data
- Financial, weather and device data are all best handled as time series

Time series in the field of Data Science



Course overview

- Chapter 1: Getting started and personalizing your first time series plot
- Chapter 2: Summarizing and describing time series data
- Chapter 3: Advanced time series analysis
- Chapter 4: Working with multiple time series
- Chapter 5: Case Study

Reading data with Pandas

```
import pandas as pd
df = pd.read_csv('ch2_co2_levels.csv')
print(df)
```

```
      datestamp    co2
0     1958-03-29  316.1
1     1958-04-05  317.3
2     1958-04-12  317.6
...
...
...
2281  2001-12-15  371.2
2282  2001-12-22  371.3
2283  2001-12-29  371.5
```

Preview data with Pandas

```
print(df.head(n=5))
```

```
    datestamp    co2
0  1958-03-29  316.1
1  1958-04-05  317.3
2  1958-04-12  317.6
3  1958-04-19  317.5
4  1958-04-26  316.4
```

```
print(df.tail(n=5))
```

```
       datestamp    co2
2279  2001-12-01  370.3
2280  2001-12-08  370.8
2281  2001-12-15  371.2
2282  2001-12-22  371.3
2283  2001-12-29  371.5
```

Check data types with Pandas

```
print(df.dtypes)
```

check the type of each column in a DataFrame

```
datestamp     object
co2          float64
dtype: object
```

Working with dates

To work with time series data in `pandas`, your date columns needs to be of the `datetime64` type.

```
pd.to_datetime(['2009/07/31', 'test'])
```

```
ValueError: Unknown string format
```

```
pd.to_datetime(['2009/07/31', 'test'], errors='coerce')
```

override
return a NaT timestamp when the object cannot be parsed

```
DatetimeIndex(['2009-07-31', 'NaT'],
    dtype='datetime64[ns]', freq=None)
```

Let's get started!

VISUALIZING TIME SERIES DATA IN PYTHON

Plot your first time series

VISUALIZING TIME SERIES DATA IN PYTHON

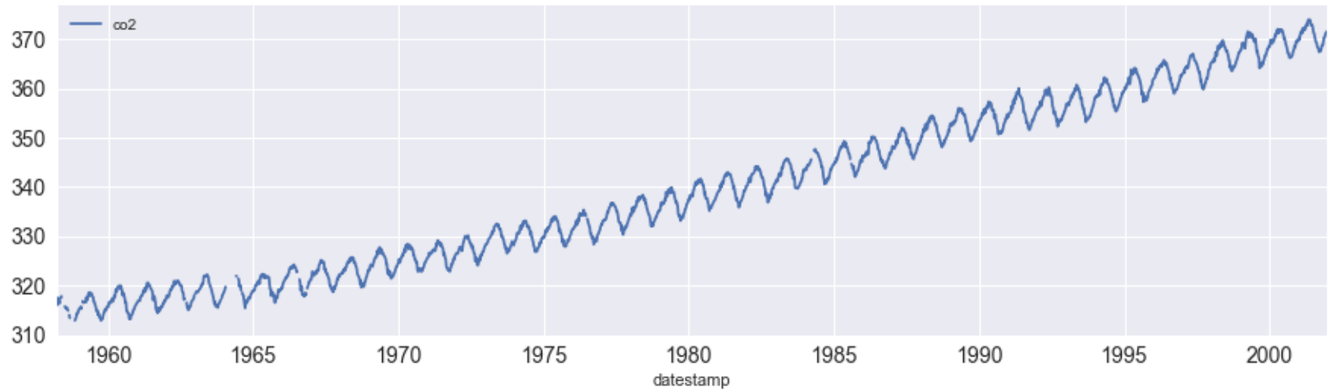
Thomas Vincent
Head of Data Science, Getty Images

The Matplotlib library

- In Python, matplotlib is an extensive package used to plot data
- The pyplot submodule of matplotlib is traditionally imported using the `plt` alias

```
import matplotlib.pyplot as plt
```

Plotting time series data



Plotting time series data

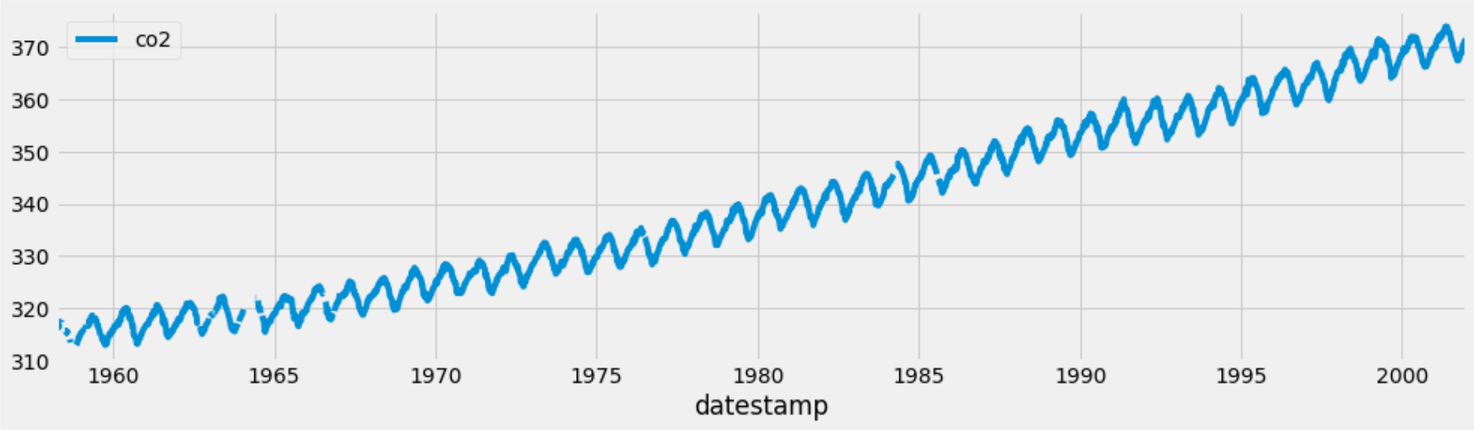
```
import matplotlib.pyplot as plt
import pandas as pd

df = df.set_index('date_column')
df.plot()
plt.show()
```

Adding style to your plots

```
plt.style.use('fivethirtyeight')
df.plot()
plt.show()
```

FiveThirtyEight style



Matplotlib style sheets

```
print(plt.style.available)
```

```
['seaborn-dark-palette', 'seaborn-darkgrid',
'seaborn-dark', 'seaborn-notebook',
'seaborn-pastel', 'seaborn-white',
'classic', 'ggplot', 'grayscale',
'dark_background', 'seaborn-poster',
'seaborn-muted', 'seaborn', 'bmh',
'seaborn-paper', 'seaborn-whitegrid',
'seaborn-bright', 'seaborn-talk',
'fivethirtyeight', 'seaborn-colorblind',
'seaborn-deep', 'seaborn-ticks']
```

Describing your graphs with labels

```
ax = df.plot(color='blue')
```

```
ax.set_xlabel('Date')
ax.set_ylabel('The values of my Y axis')
ax.set_title('The title of my plot')
plt.show()
```

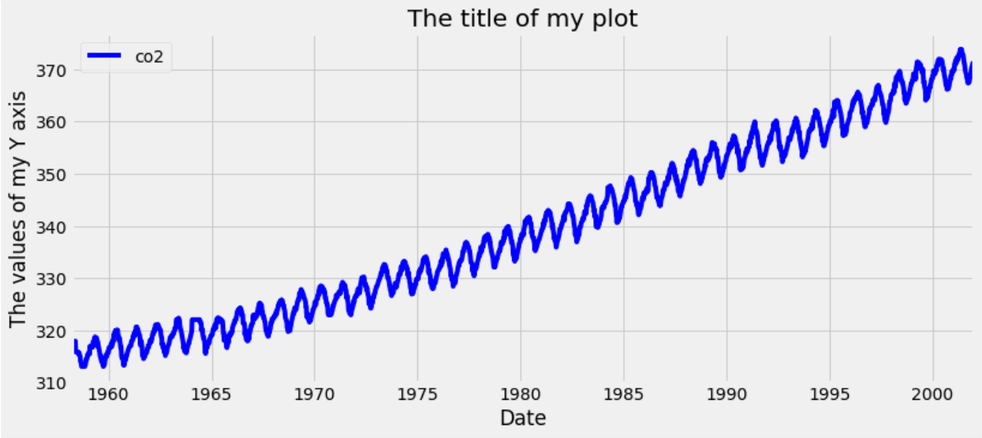
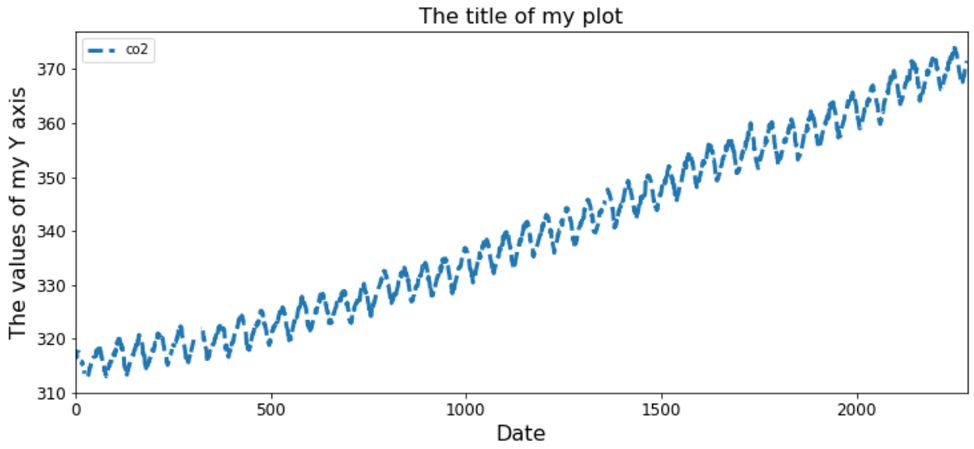


Figure size, linewidth, linestyle and fontsize

```
ax = df.plot(figsize=(12, 5), fontsize=12,
             linewidth=3, linestyle='--')
ax.set_xlabel('Date', fontsize=16)
ax.set_ylabel('The values of my Y axis', fontsize=16)
ax.set_title('The title of my plot', fontsize=16)
plt.show()
```



Let's practice!

VISUALIZING TIME SERIES DATA IN PYTHON

Customize your time series plot

VISUALIZING TIME SERIES DATA IN PYTHON

Thomas Vincent
Head of Data Science, Getty Images

Slicing time series data

```
discoveries['1960':'1970']
```

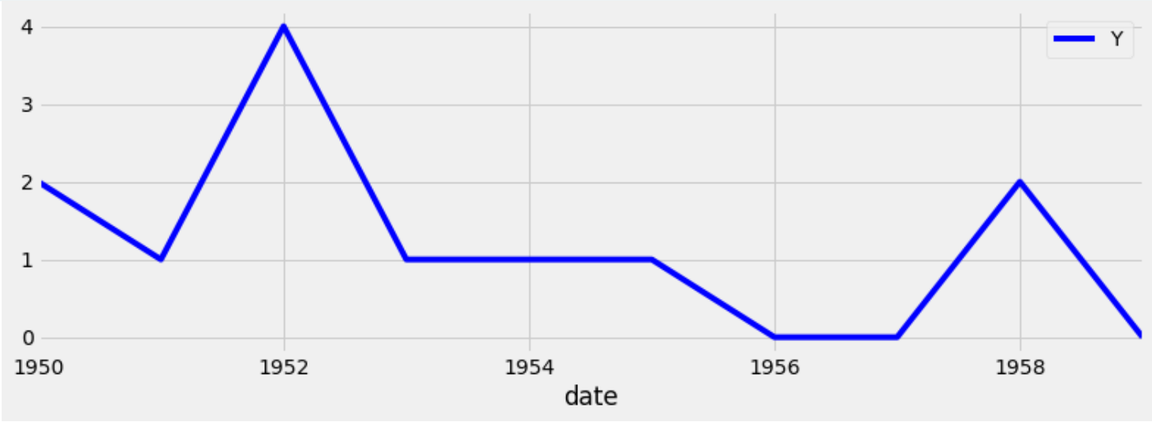
```
discoveries['1950-01':'1950-12']
```

```
discoveries['1960-01-01':'1960-01-15']
```

Plotting subset of your time series data

```
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')
df_subset = discoveries['1960':'1970']
```

```
ax = df_subset.plot(color='blue', fontsize=14)
plt.show()
```



Adding markers

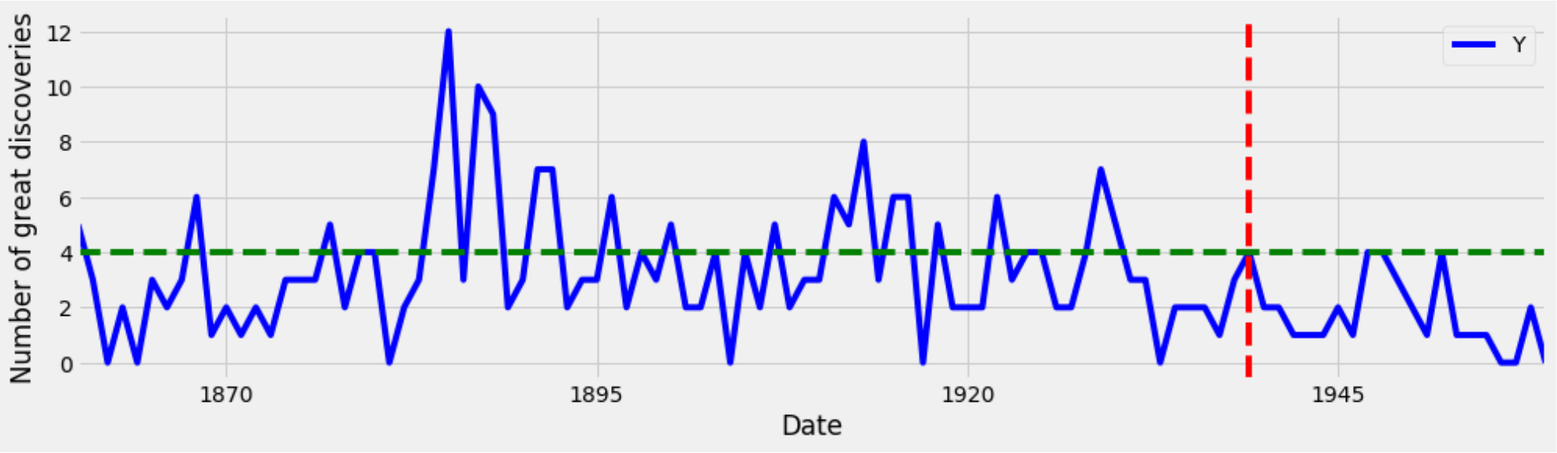
```
ax.axvline(x='1969-01-01',
           color='red',
           linestyle='--')
```

```
ax.axhline(y=100,
           color='green',
           linestyle='--')
```

Using markers: the full code

```
ax = discoveries.plot(color='blue')
ax.set_xlabel('Date')
ax.set_ylabel('Number of great discoveries')

ax.axvline('1969-01-01', color='red', linestyle='--')
ax.axhline(4, color='green', linestyle='--')
```



Highlighting regions of interest

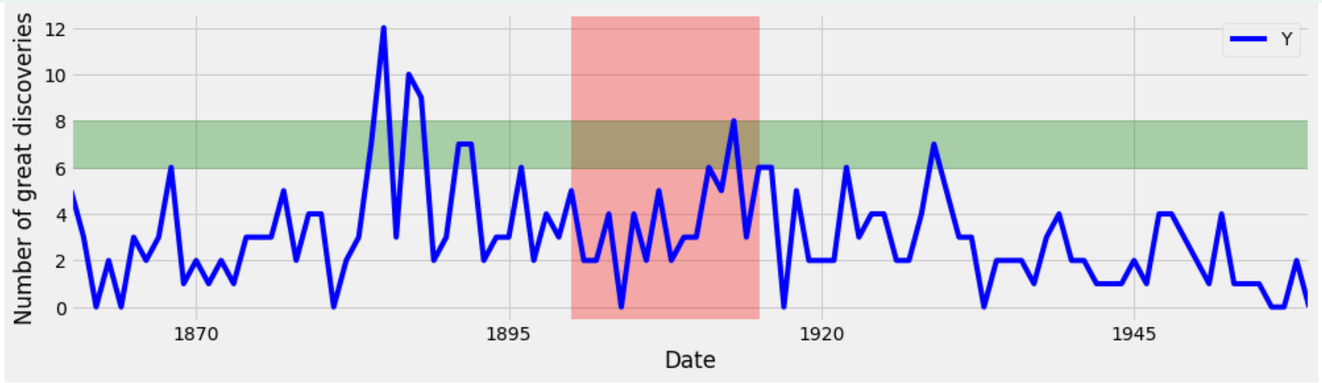
```
ax.axvspan('1964-01-01', '1968-01-01',
           color='red', alpha=0.5)
```

```
ax.axhspan(8, 6, color='green',
           alpha=0.2)
```

Highlighting regions of interest: the full code

```
ax = discoveries.plot(color='blue')
ax.set_xlabel('Date')
ax.set_ylabel('Number of great discoveries')
```

```
ax.axvspan('1964-01-01', '1968-01-01', color='red',
alpha=0.3)
ax.axhspan(8, 6, color='green', alpha=0.3)
```



Let's practice!

VISUALIZING TIME SERIES DATA IN PYTHON