

Part 2 — Solutions for Lecture 5

TECH2: Introduction to Programming, Data, and Information Technology

Richard Foltyn

Norwegian School of Economics (NHH)

October 30, 2024

See GitHub repository for notebooks and data:

<https://github.com/richardfoltyn/TECH2-H24>

Contents

1 Plotting with Matplotlib	1
1.1 Line plots	1
1.2 Scatter plots	3
1.2.1 Part (1)	3
2 Working with multiple plots (axes)	5
3 Plotting with pandas	6

1 Plotting with Matplotlib

1.1 Line plots

Your turn. Use the data files located in the folder `../data/FRED` to perform the following tasks:

1. Load the data in `REALRATE.csv` and `FEDFUNDS.csv` into two different DataFrames. The files contain the 1-year real interest rate and the Federal Funds rate at monthly frequency.

Hint: Use `pd.read_csv(..., parse_dates=['DATE'], index_col='DATE')` to automatically parse strings stored in the `DATE` column as dates and set `DATE` as the index.

2. Plot the `REALRATE` using a blue dashed line with line width 0.5 and `FEDFUNDS` using an orange line with line width 0.75 in the same figure.

Solution.

Part (1)

```
[1]: # Path to local data folder
DATA_PATH = '../data/FRED'
DATA_PATH = '/home/richard/repos/teaching/TECH2-H24/data/FRED'

import pandas as pd
```

```
# Load both time series
realrate = pd.read_csv(f'{DATA_PATH}/REALRATE.csv', parse_dates=['DATE'], index_col='DATE')
fedfunds = pd.read_csv(f'{DATA_PATH}/FEDFUNDS.csv', parse_dates=['DATE'], index_col='DATE')
```

Part (2)

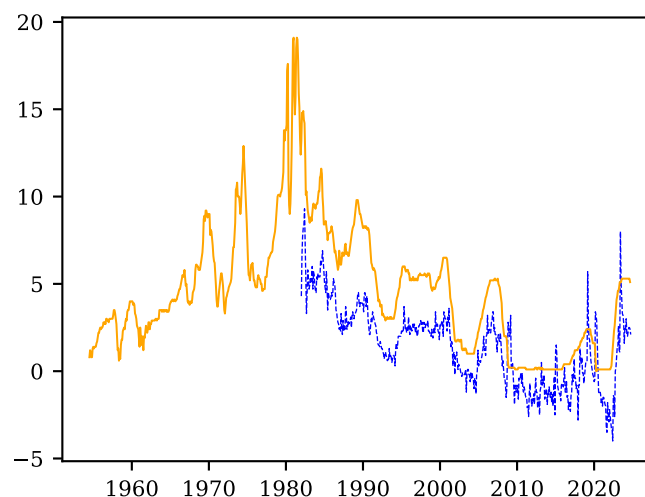
If you set the index accordingly, Matplotlib automatically displays the date on the x-axis without having to specify the x-values explicitly.

```
[2]: import matplotlib.pyplot as plt

# Plot real rate with as a dashed line with width 0.5
plt.plot(realrate, c='blue', ls='--', lw=0.5)

# Plot the Fed Funds rate as a solid orange line with width 0.75
plt.plot(fedfunds, c='orange', lw=0.75)
```

```
[2]: [<matplotlib.lines.Line2D at 0x7f90f8053e60>]
```



We could alternatively use the dates as x-values. For that, we first need to move them back to columns:

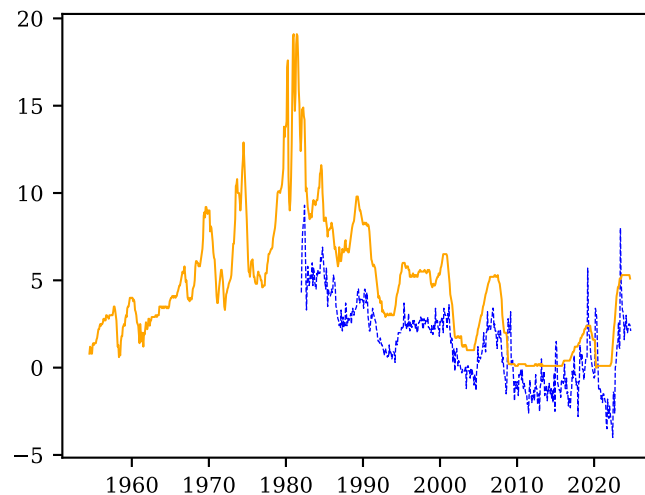
```
[3]: # Move date to columns
realrate = realrate.reset_index(drop=False)
fedfunds = fedfunds.reset_index(drop=False)
```

We can now plot both time series, specifying both the x- and y-values:

```
[4]: # Plot real rate, explicitly specifying date
plt.plot(realrate['DATE'], realrate['REALRATE'], c='blue', ls='--', lw=0.5)

# Plot the Fed Funds rate as a solid orange line with width 0.75
plt.plot(fedfunds['DATE'], fedfunds['FEDFUNDS'], c='orange', lw=0.75)
```

```
[4]: [<matplotlib.lines.Line2D at 0x7f90ecc53440>]
```



1.2 Scatter plots

Your turn. Use the data files located in the folder `../data/FRED` to perform the following tasks:

1. Load the data in `REALRATE.csv` and `FEDFUNDS.csv` into two different DataFrames. The files contain the 1-year real interest rate and the Federal Funds rate at monthly frequency.

Hint: Use `pd.read_csv(..., parse_dates=['DATE'], index_col='DATE')` to automatically parse strings stored in the `DATE` column as dates and set `DATE` as the index.

2. Create a scatter plot of `REALRATE` (on the y-axis) against `FEDFUNDS` (on the x-axis). Specify the arguments `edgecolors='blue'` and `color='none'` to plot the data as blue rings.

Hint: `scatter()` requires the arguments to be of equal length, which is not the case here. Restrict the data to the same time period, for example by using `join(..., how='inner')`.

Solution.

1.2.1 Part (1)

```
[5]: # Path to local data folder
DATA_PATH = '../data/FRED'
DATA_PATH = '/home/richard/repos/teaching/TECH2-H24/data/FRED'

import pandas as pd

# Load both time series
realrate = pd.read_csv(f'{DATA_PATH}/REALRATE.csv', parse_dates=['DATE'], index_col='DATE')
fedfunds = pd.read_csv(f'{DATA_PATH}/FEDFUNDS.csv', parse_dates=['DATE'], index_col='DATE')
```

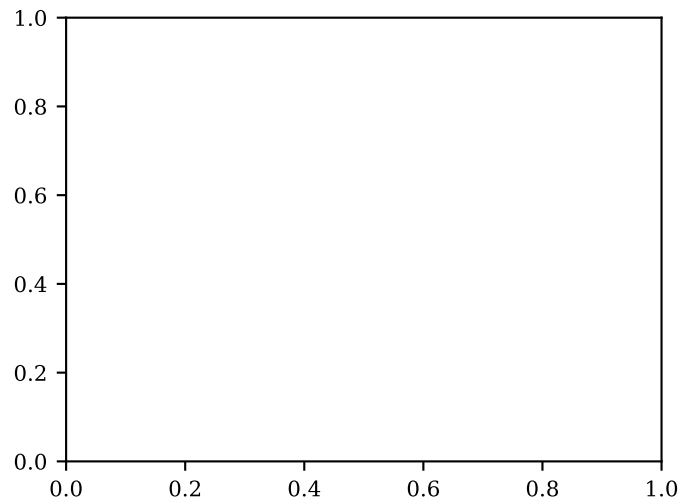
Part (2)

Trying to create the scatter plot on the original data results in the following error:

```
[6]: import matplotlib.pyplot as plt
```

```
plt.scatter(realrate, fedfunds, edgecolors='blue', color='none')
```

```
ValueError: x and y must be the same size
```



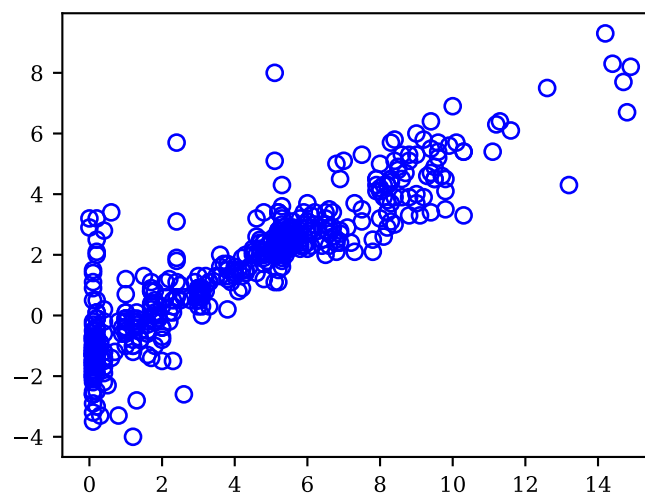
This error arises because these variables are observed for different time periods. One way to rectify this is to merge them into a single DataFrame using `join(..., how='inner')` which only keeps periods for which both variables are observed.

```
[ ]: # Perform inner join to keep only intersection of dates  
df = realrate.join(fedfunds, how='inner')
```

Once we have harmonized the number of observations, we can create the scatter plot as follows:

```
[ ]: plt.scatter(df['FEDFUNDS'], df['REALRATE'], edgecolors='blue', color='none')  
plt.xlabel('Fed funds rate')  
plt.ylabel('Real interest rate')
```

```
[ ]: <matplotlib.collections.PathCollection at 0x7f05dc674ef0>
```



2 Working with multiple plots (axes)

Your turn. Create a figure with 3 columns (on a single row) and plot the following functions on the interval $[0, 6]$:

- Subplot 1: $y = \sin(x)$
- Subplot 2: $y = \sin(2 * x)$
- Subplot 3: $y = \sin(4 * x)$

Hint: The sine function can be imported from NumPy as `np.sin()`.

Solution.

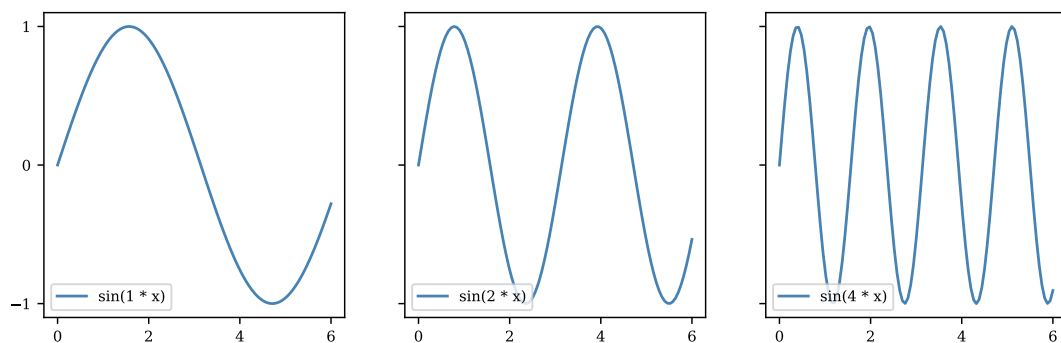
```
[ ]: import matplotlib.pyplot as plt
import numpy as np

# Create a figure with 3 subplots arranged in 3 columns
fig, axes = plt.subplots(1, 3, sharex=True, sharey=True, figsize=(10, 3))

# Define common x-values for all subplots
xvalues = np.linspace(0, 6, 101)

# Iterate through axes
for i, ax in enumerate(axes):
    # Create plot-specific y-values
    yvalues = np.sin(2**i * xvalues)
    ax.plot(xvalues, yvalues, color='steelblue', lw=1.5, label=f'sin({2**i} * x)')
    ax.set_yticks([-1, 0, 1])

# Place legend in lower left corner
ax.legend(loc='lower left')
```



3 Plotting with pandas

Your turn. Use the data files located in the folder ../data to perform the following tasks:

1. Load the macroeconomic time series data from FRED.csv.

Hint: Use `pd.read_csv(..., parse_dates=['DATE'])` to automatically set parse strings as dates.

2. Create a line plot, showing both the unemployment rate UNRATE and the inflation rate INFLATION in a single graph.

Solution.

Part (1)

```
[ ]: # Path to local data/ folder
DATA_PATH = '../data/FRED'
DATA_PATH = '/home/richard/repos/teaching/TECH2-H24/data/FRED'

# Path to FRED.csv;
filepath = f'{DATA_PATH}/FRED_monthly_all.csv'

# Read CSV data
df = pd.read_csv(filepath, parse_dates=['DATE'])
```

Part (2)

```
[ ]: # Plot unemployment rate and inflation
df.plot(x='DATE', y=['UNRATE', 'INFLATION'], ylabel='Percent')

[ ]: <Axes: xlabel='DATE', ylabel='Percent %'>
```

