

Using merge_asof() to study stocks - Corrected

The screenshot shows a web browser window displaying a DataCamp course page titled "Using merge_asof() to study stocks". The page is divided into two main sections: "Instructions" on the left and a code editor on the right. The "Instructions" section contains a list of steps: 1. Use merge_asof() to merge jpm (left table) and wells together on the date_time column, where the rows with the nearest times are matched, and with suffixes ('_jpm', '_wells'). Save to jpm_wells. 2. Use merge_asof() to merge jpm_wells (left table) and bac together on the date_time column, where the rows with the closest times are matched, and with suffixes ('_jpm', '_bac'). Save to jpm_wells_bac. 3. Plot the close prices of jpm, wells, and bac from jpm_wells_bac. The code editor on the right shows the following Python code:

```
1 # Use merge_asof() to merge jpm and wells
2 jpm_wells = pd.merge_asof(
3     jpm, wells,
4     on='date_time',
5     direction='nearest',
6     suffixes=('_jpm', '_wells')
7 )
8
9 # Compute price diff
10 price_diffs = jpm_wells_bac.diff()
11
12 # Plot the price diff of the close of jpm, wells and bac only
13 price_diffs.plot()
14 plt.show()
```

The code editor also has a "Run Code" button and a "Submit Answer" button. The bottom of the browser window shows the Windows taskbar with the date 04/12/2024 and time 12:46.

Question:

Use merge_asof() to merge jpm and wells together on the date_time column, where the rows with the nearest times are matched, and with suffixes ('_jpm', '_wells'). Save to jpm_wells. Then use merge_asof() to merge jpm_wells and bac together on the date_time column, where the rows with the closest times are matched, and with suffixes ('_jpm', '_bac'). Save to jpm_wells_bac. Finally, compute the price change using the .diff() method and plot the close prices of jpm, wells, and bac from the resulting dataframe.

Answer:

```
# Use merge_asof() to merge jpm and wells with direction='nearest'
```

```
jpm_wells = pd.merge_asof(
    jpm, wells,
    on='date_time',
    suffixes=('_', '_wells'),
    direction='nearest'
)
```

```
# Use merge_asof() to merge jpm_wells and bac with direction='nearest'
```

```
jpm_wells_bac = pd.merge_asof(
    jpm_wells, bac,
    on='date_time',
```

```

    suffixes=('_jpm', '_bac'),
    direction='nearest'
)

# Compute price differences
price_diffs = jpm_wells_bac.diff()

# Plot the price differences of close_jpm, close_wells, and close_bac
price_diffs.plot(
    y=['close_jpm', 'close_wells', 'close_bac']
)
plt.show()

```

Code Explanation:

1. `jpm_wells = pd.merge_asof(..., direction='nearest')`:

This line merges the `jpm` and `wells` dataframes on the `'date_time'` column using the `'merge_asof'` function with the `direction` parameter set to `'nearest'`. This ensures that rows with the nearest timestamp are matched, and suffixes are applied to differentiate columns.

2. `jpm_wells_bac = pd.merge_asof(..., direction='nearest')`:

This line merges the `jpm_wells` dataframe with the `bac` dataframe on the `'date_time'` column using the `'merge_asof'` function with `direction='nearest'`. This combines the three dataframes by aligning the nearest timestamps.

3. `price_diffs = jpm_wells_bac.diff()`:

This line computes the difference between consecutive rows for each column in the dataframe, calculating the changes in close prices over time.

4. `price_diffs.plot(...)`:

This line creates a plot of the close prices of `jpm`, `wells`, and `bac`. The `'y'` parameter specifies the columns to plot, and `plt.show()` displays the visualization.