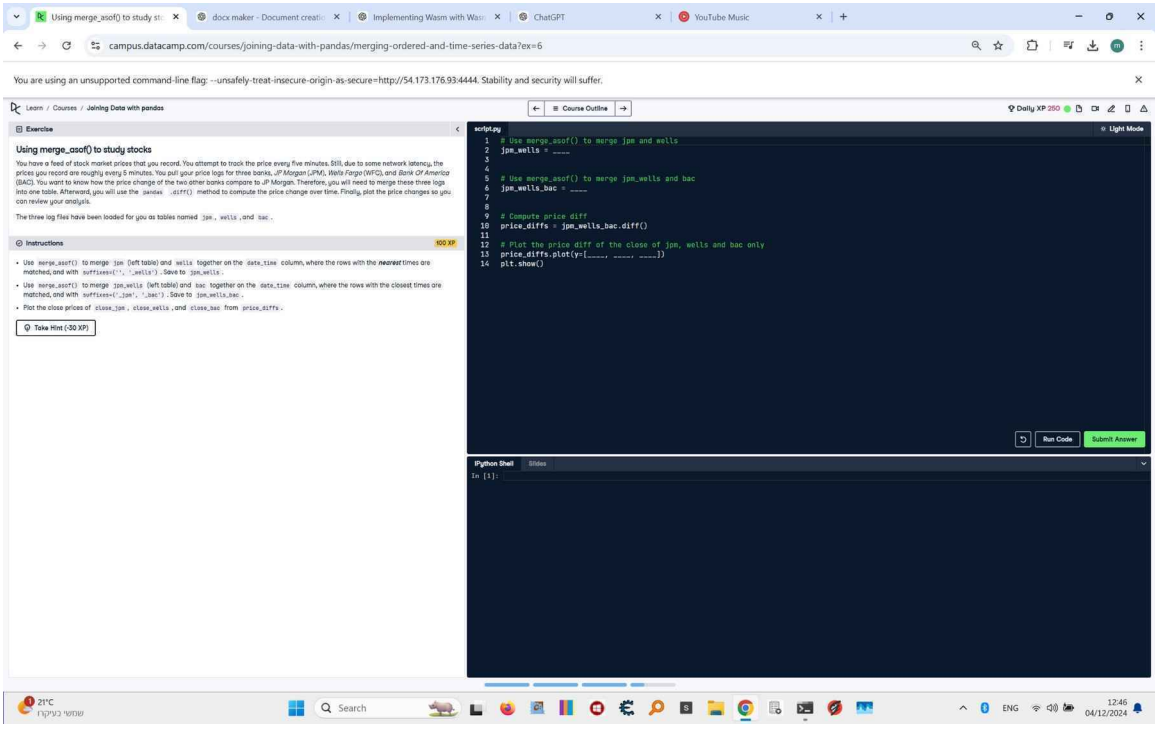
Using merge_asof() to study stocks - Corrected



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.