**Data Visualisation - Unemployment: How to use Grids**

For the next challenge, carry over your existing code once again (by copy-pasting the entire cell) and make some modifications.

Challenge

Plot the search for "unemployment benefits" against the official unemployment rate.

1. Change the title to: Monthly Search of "Unemployment Benefits" in the U.S. vs the U/E Rate
2. Change the y-axis label to: FRED U/E Rate
3. Change the axis limits
4. Add a grey [grid](https://matplotlib.org/3.2.1/api/_as_gen/matplotlib.pyplot.grid.html) to the chart to better see the years and the U/E rate values. Use dashed lines for the line style.
5. Can you discern any seasonality in the searches? Is there a pattern?

.

.

..

...

..

.

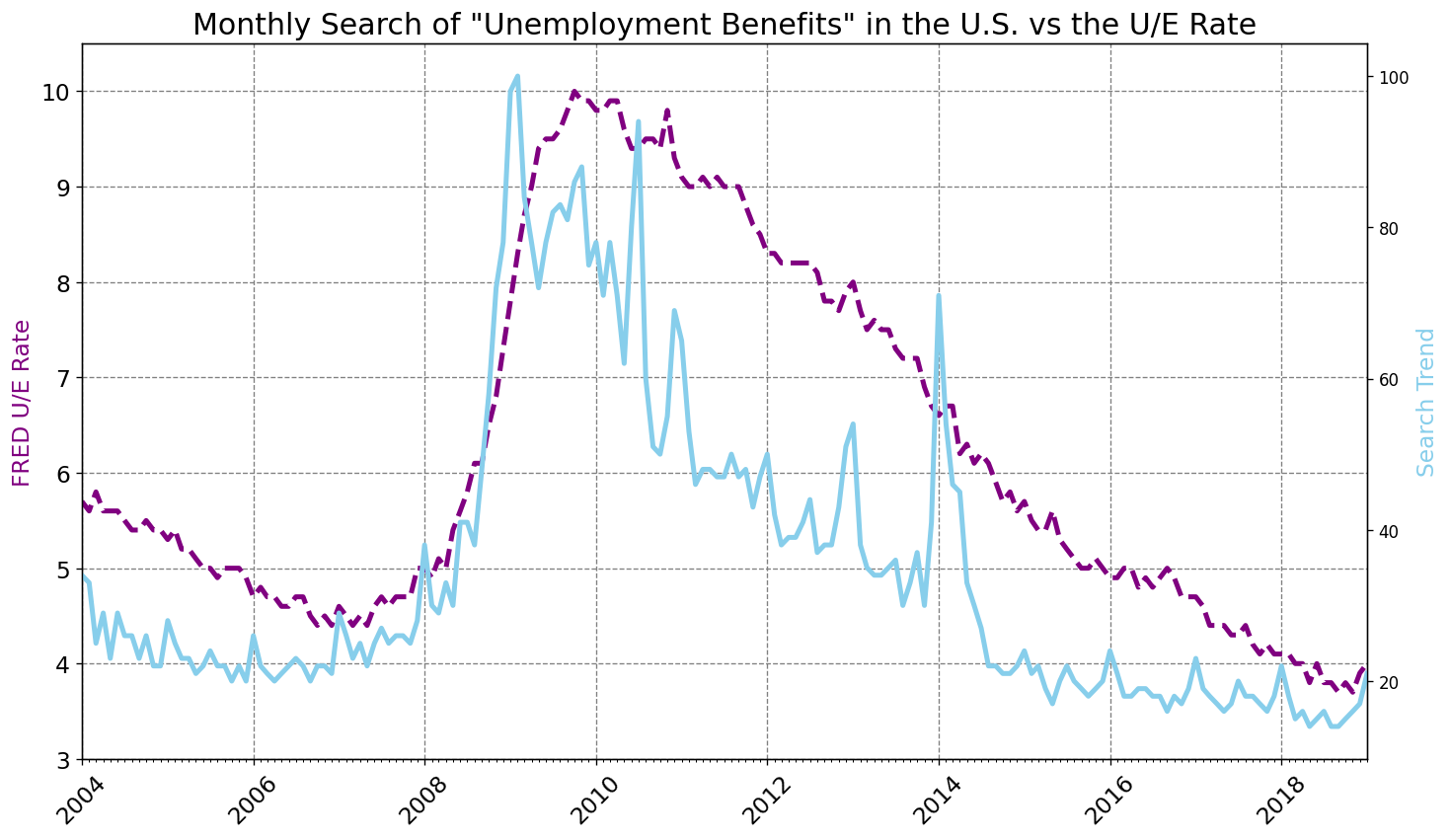
.

**Solution: Adding a grid to spot seasonality**

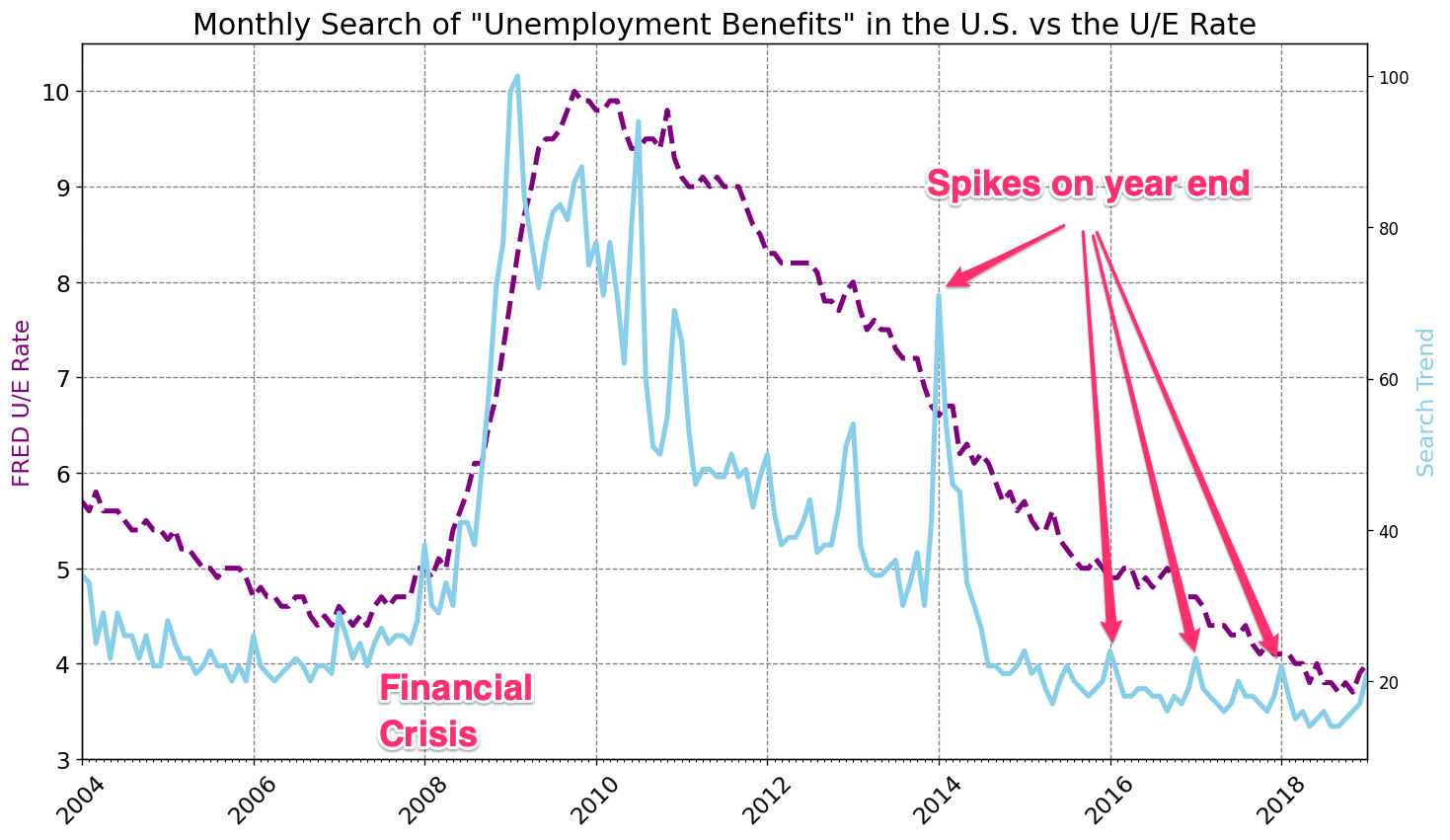
Ok, so there are relatively few changes you had to make here. Just the labels and the dataset we're using. The line of code I wanted you to figure out from the documentation was this one:

1. ax1.grid(color='grey', linestyle='--')

This overlays a grid of dashed lines, so that we get the following look:



Notice how we can now clearly see the vertical dashed lines line up with spikes in searches for "Unemployment benefits". Many of the spikes are at year-end - in December. This clearly shows that there is seasonality in the job market. What else do we see? We see that the financial crisis in 2007/2008 caused a massive spike in unemployment. It took around 10 years (2007-2017) for the unemployment to reach the same level it had before the crisis.



Interestingly the big spike in searches for Unemployment benefits at the end of 2013 was not accompanied by a big increase in the unemployment rate. Something else must have been going on around that time.

Here's the full code for the cell:

1. plt.figure(figsize=(14,8), dpi=120)
2. plt.title('Monthly Search of "Unemployment Benefits" in the U.S. vs the U/E Rate', fontsize=18)
3. plt.yticks(fontsize=14)
4. plt.xticks(fontsize=14, rotation=45)
6. ax1 = plt.gca()
7. ax2 = ax1.twinx()
9. ax1.set\_ylabel('FRED U/E Rate', color='purple', fontsize=14)
10. ax2.set\_ylabel('Search Trend', color='skyblue', fontsize=14)
12. ax1.xaxis.set\_major\_locator(years)
13. ax1.xaxis.set\_major\_formatter(years\_fmt)
14. ax1.xaxis.set\_minor\_locator(months)
16. ax1.set\_ylim(bottom=3, top=10.5)
17. ax1.set\_xlim([df\_unemployment.MONTH.min(), df\_unemployment.MONTH.max()])
19. # Show the grid lines as dark grey lines
20. ax1.grid(color='grey', linestyle='--')
22. # Change the dataset used
23. ax1.plot(df\_unemployment.MONTH, df\_unemployment.UNRATE,
24. color='purple', linewidth=3, linestyle='--')
25. ax2.plot(df\_unemployment.MONTH, df\_unemployment.UE\_BENEFITS\_WEB\_SEARCH,
26. color='skyblue', linewidth=3)
28. plt.show()

The search volume moves around quite a bit - month on month. Perhaps we can smooth out the search volumes to get a slightly different picture (pun intended!).

Challenge

Calculate the 3-month or 6-month rolling average for the web searches. Plot the 6-month rolling average search data against the actual unemployment. What do you see? Which line moves first?

*Hint*: Take a look at our prior lesson on Programming Languages where we smoothed out time-series data.

.

.

..

...

..

.

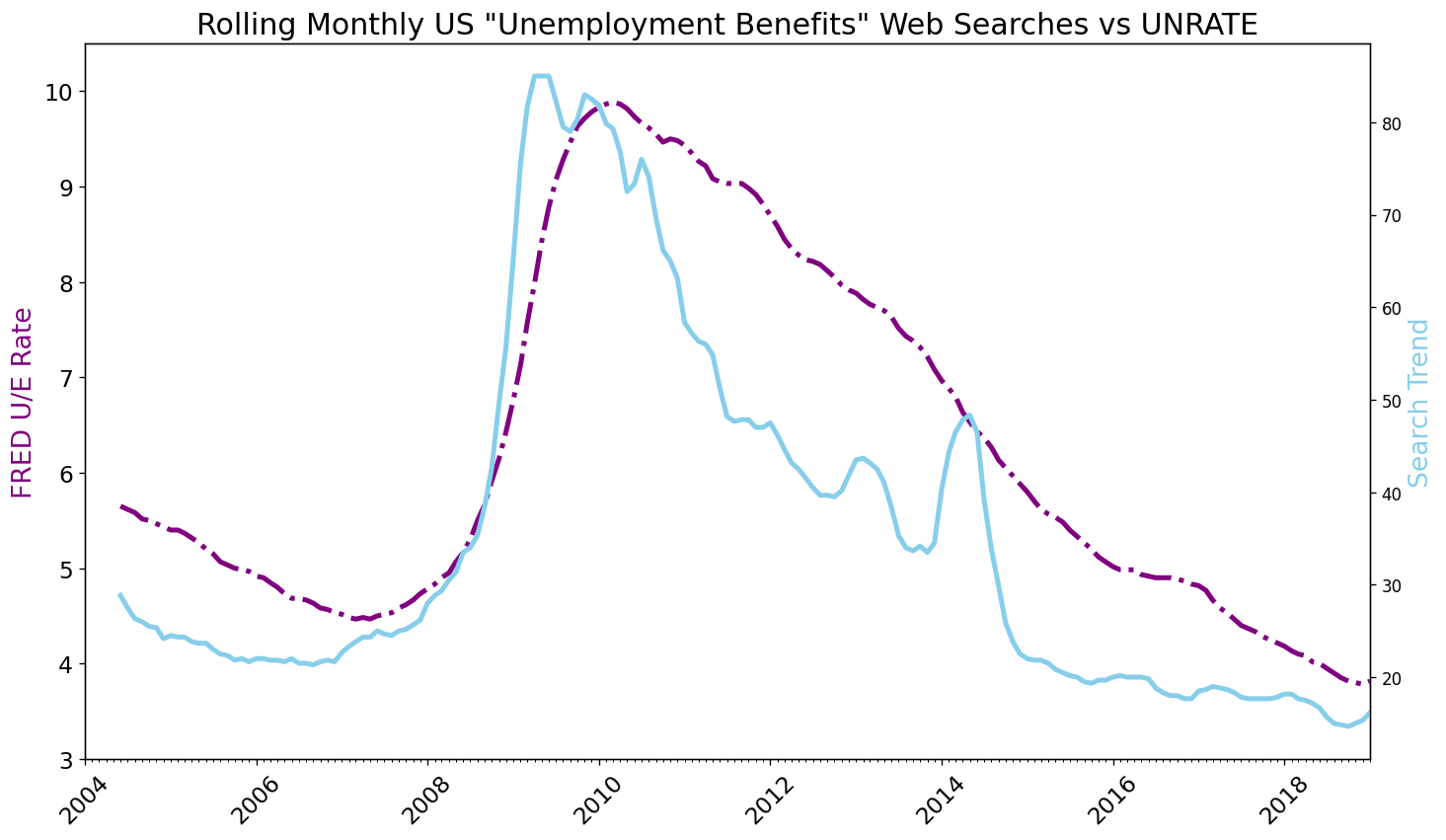
.

**Solution: Rolling Average**

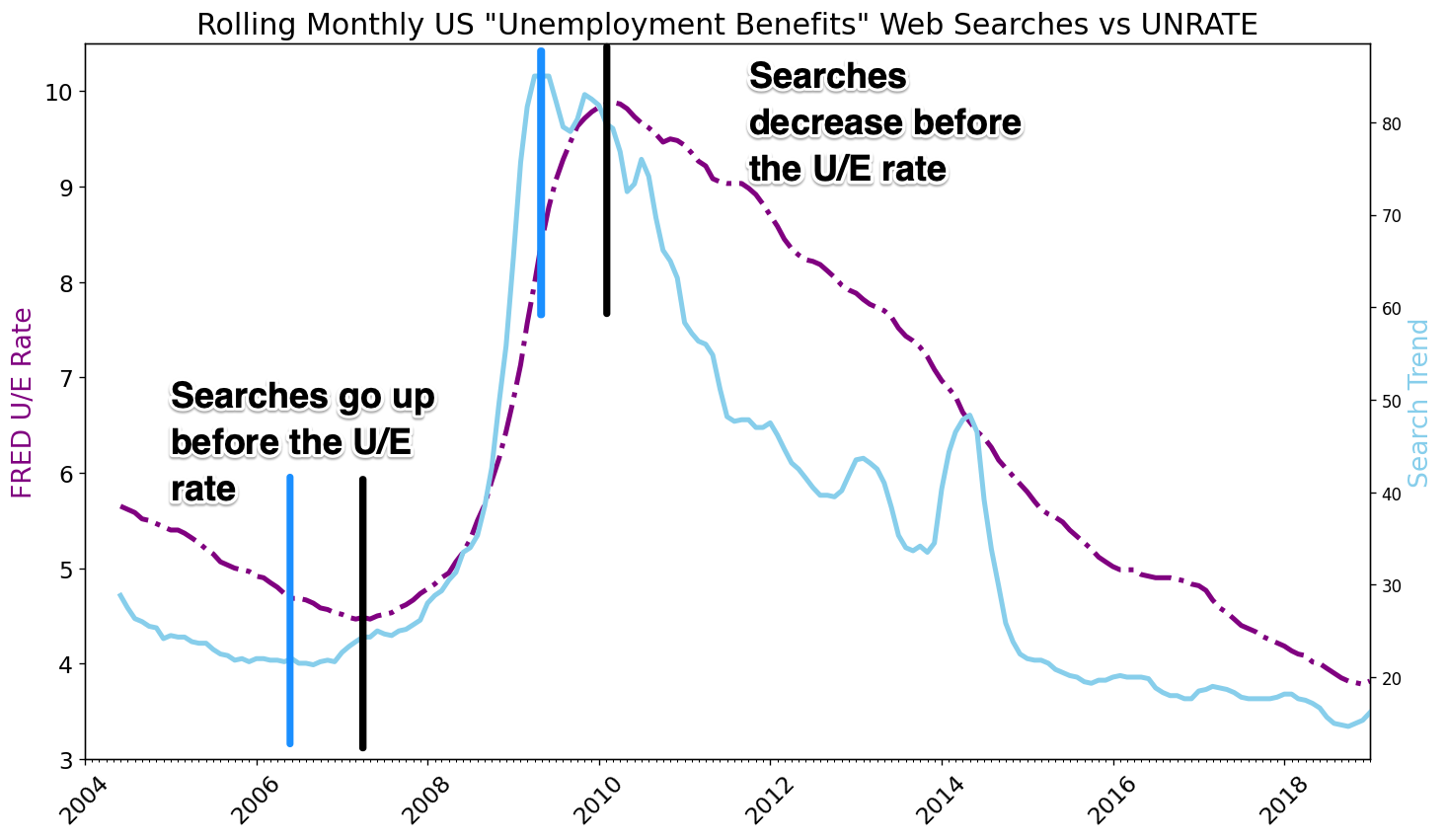
You can create a rolling average using .rolling() and .mean() functions together.

1. roll\_df = df\_unemployment[['UE\_BENEFITS\_WEB\_SEARCH', 'UNRATE']].rolling(window=6).mean()

Your plot should look something like this:



What is this telling us? We see that searches for "Unemployment Benefits" happen before the actual official unemployment rate goes up. Similarly, the search popularity for the term goes down before the unemployment rate decreases. In other words, these searches seem to act as a leading economic indicator for the unemployment rate (which is a lagging indicator).



Here's the full code for the cell:

1. plt.figure(figsize=(14,8), dpi=120)
2. plt.title('Rolling Monthly US "Unemployment Benefits" Web Searches vs UNRATE', fontsize=18)
3. plt.yticks(fontsize=14)
4. plt.xticks(fontsize=14, rotation=45)
6. ax1 = plt.gca()
7. ax2 = ax1.twinx()
9. ax1.xaxis.set\_major\_locator(years)
10. ax1.xaxis.set\_major\_formatter(years\_fmt)
11. ax1.xaxis.set\_minor\_locator(months)
13. ax1.set\_ylabel('FRED U/E Rate', color='purple', fontsize=16)
14. ax2.set\_ylabel('Search Trend', color='skyblue', fontsize=16)
16. ax1.set\_ylim(bottom=3, top=10.5)
17. ax1.set\_xlim([df\_unemployment.MONTH[0], df\_unemployment.MONTH.max()])
19. # Calculate the rolling average over a 6 month window
20. roll\_df = df\_unemployment[['UE\_BENEFITS\_WEB\_SEARCH', 'UNRATE']].rolling(window=6).mean()
22. ax1.plot(df\_unemployment.MONTH, roll\_df.UNRATE, 'purple', linewidth=3, linestyle='-.')
23. ax2.plot(df\_unemployment.MONTH, roll\_df.UE\_BENEFITS\_WEB\_SEARCH, 'skyblue', linewidth=3)
25. plt.show()