Hi Sunny –

I think its fantastic that we’re able to work with the amcharts library and I’m very excited about the ability to dynamically support the charts with data we have hosted in the background in postgres.

The graph overall is great. I have a number of design tweaks I’ve outlined below that I’m hoping to have implemented. Then, I’ve outlined a description for the second graphic. For now, we can just have both render on the same Django site. We can transition to website design after we’ve finished with the graphics.

Graph #1 edits

Primary theme color:

I decided that I prefer the lighter color theme over the current darker version. Eventually, I’d like to accompany the charts with a lot of text and I think lighter will end up being better for the overall purpose. Can we go back to the default theme? I’ve highlighted it below:

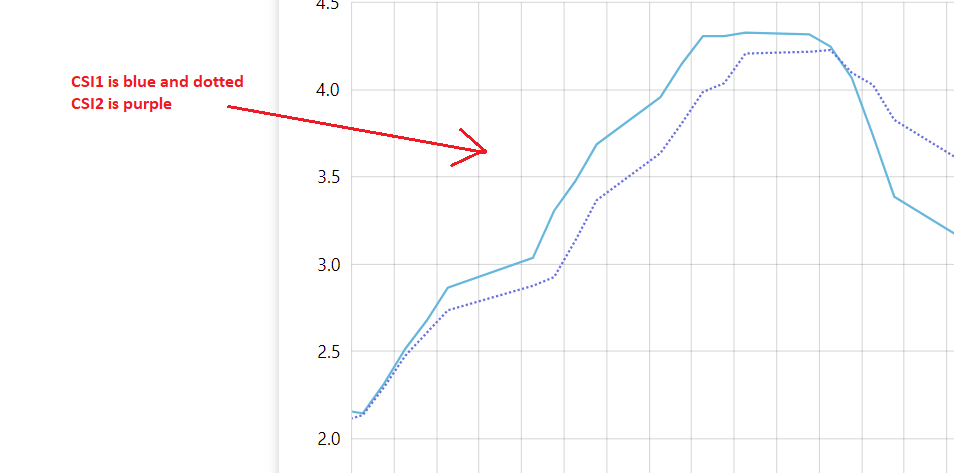


Colors of lines and axes:

I like how customizable the charts are… if possible, I’d like to adjust the colors of the lines/axes in a way that may make it easier to differentiate between the items presented.

Can we make the following adjustments?

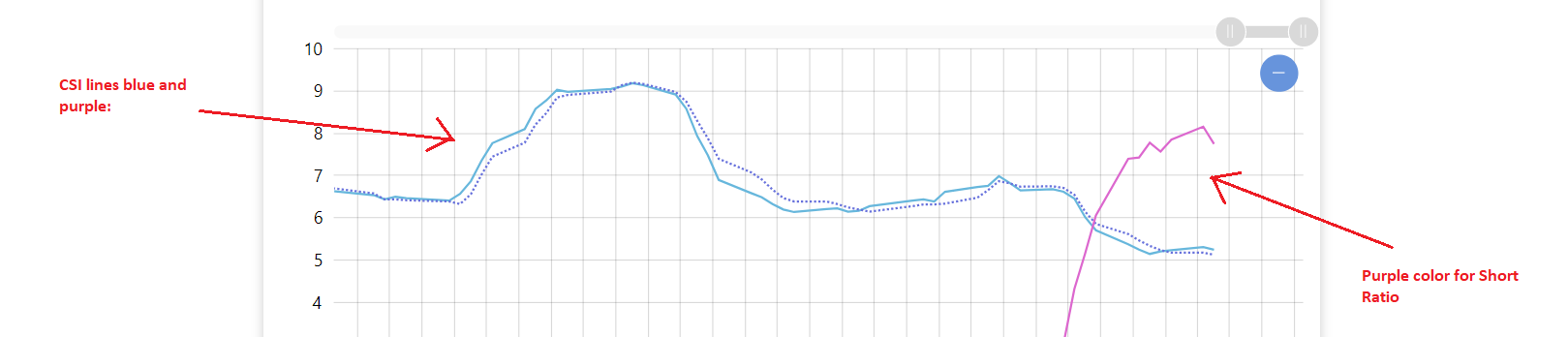
CSI lines 1 & 2 – Can these lines be changed to the same blue/purple format as seen below (same as my current temporary site):



SPY – No change here

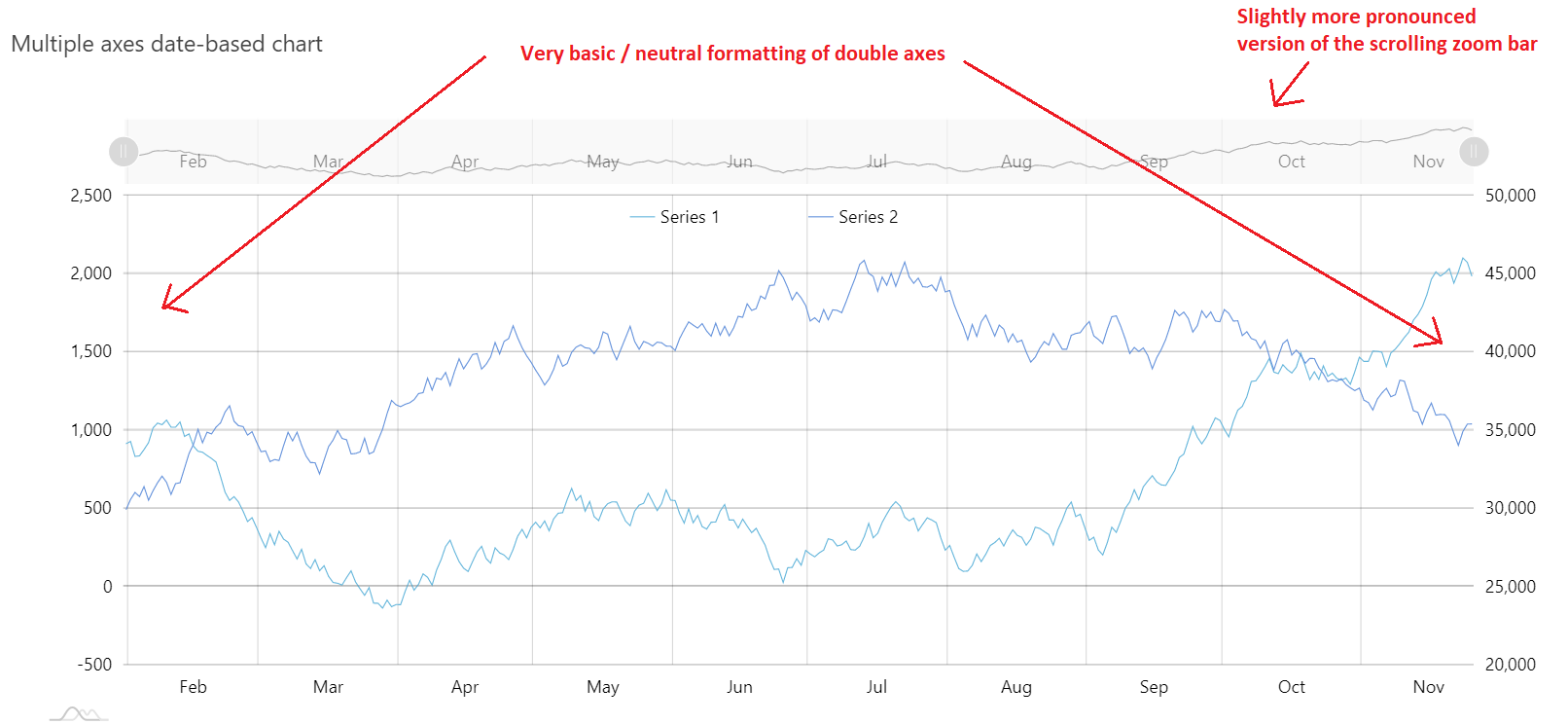
Short Ratio –

Can the color for Short Ratio be changed to the deeper purple shown below:



Axes formatting:

I’d like to see if we can simplify the axes colors a bit. Can we try and mimic one of the more basic examples shown on the amcharts website? I took a quick screenshot of one below… in this version, all the axes are the same format. I feel the current use of multiple colors distracts from the lines on the chart a bit.



Zoom function:

Is it possible to incorporate the above example’s zoom function? This particular example is a bit more pronounced than the one we’re using currently, and I feel it may lead to an improved user experience.

Axes label names:

Can the labels of the axes be changed to the following:

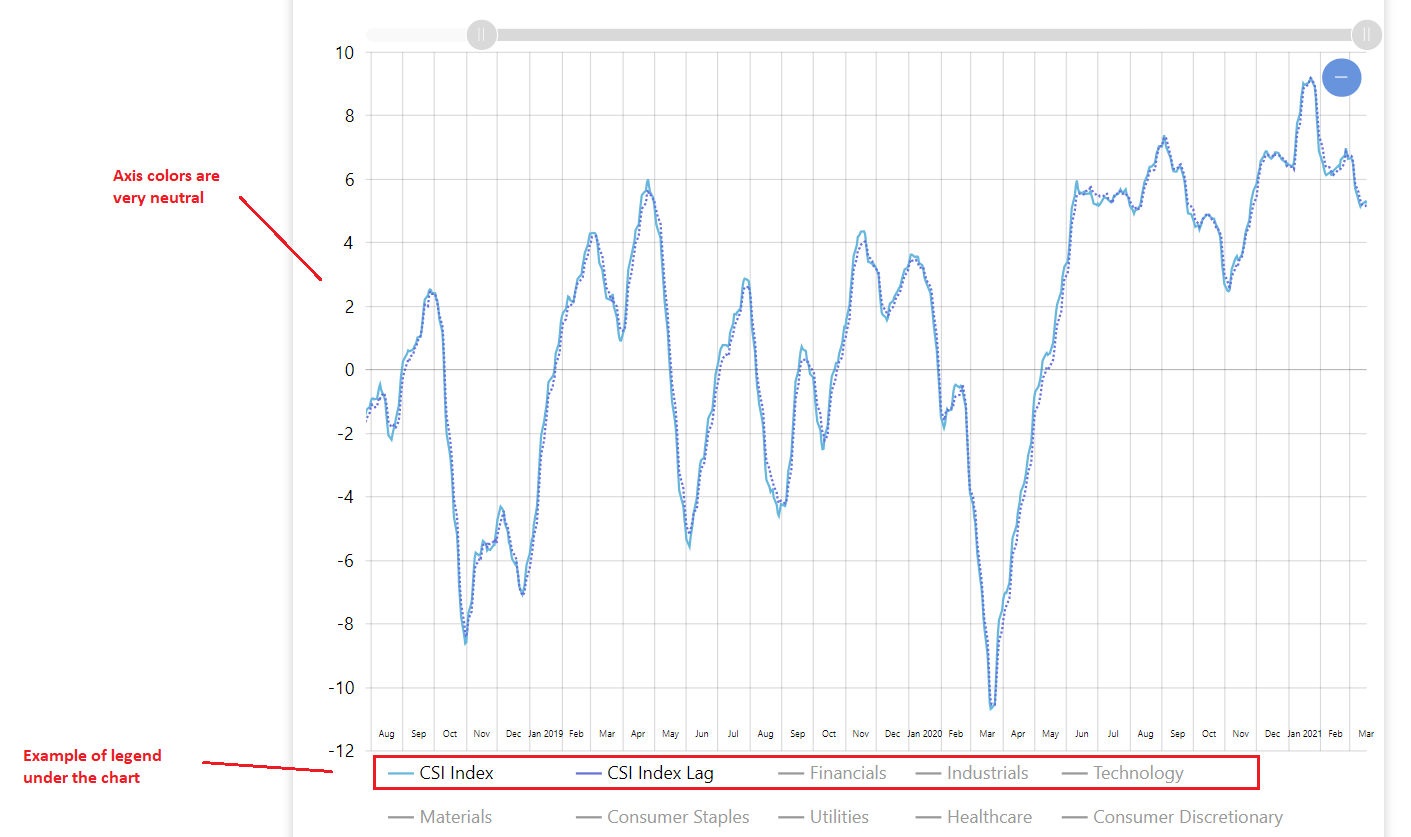
Cyclical Strength Index

S&P 500

Short Ratio

Legend location:

I know I had originally said I preferred the legend on the right side of the graph… but I believe this was an error in judgement on my part. Can we move the legend under the chart instead? I was experimenting with the chart and rendering it in a mobile version, and having the legend under the chart will be better for a horizontal screen such as a phone. The quick screenshot below shows an example from this (taken from my current temp site).



Data to display when chart first loads:

When the chart first loads, can only the data for CSI and SPY be displayed? Short Ratio data can then be displayed if the user activates it by clicking “Short Ratio” on the legend.

Short Ratio skewed at beginning – allow upload of NaNs

So this was an issue I was facing on my end when trying to upload the data to postgres. Do you mind taking a look at the attached CSV and try uploading it to postgres? (to replace the current data)

Postgres was giving errors when I tried to upload any rows that contained NaN values. I had to use .fillna(0) to replace the NaN values with 0s. Unfortunately though, this has resulted in the graphed output of Short Ratio being very skewed in the beginning… and it impacts the axis distribution quite a bit when zoomed out. I’m hoping that allowing the uploading of NaN values for dates where there’s no Short Ratio data will fix the issue.

Version of Short Ratio to graph:

Right now in postgres, there’s two versions of short ratio data. There’s short\_ratio and short\_ratio3 – Can the version graphed actually be short\_ratio3? I would prefer the labels on the graph remain as just “Short Ratio” (without the 3) even though what’s actually graphed is short\_ratio3.

Two more items to graph –

Can two more items be added to the chart? Both of these will be associated with Cyclical Strength Index axis. They are:

“1 Std Dev”

“2 Std Dev”

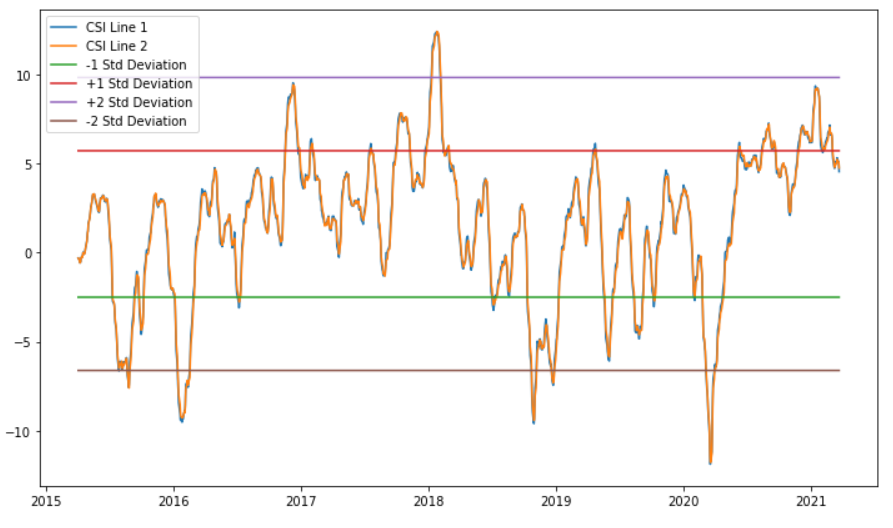
1 Std Dev will correspond to two different columns in the database table. They are: “std\_upper” and “std\_lower”. These values represent 1 standard deviation range of CSI.

2 Std Dev will be the same principle… but for “two\_std\_upper” and “two\_std\_lower”.

Consequently, clicking the 1 Std Dev button in the legend will actually trigger two lines. The purpose of these lines is to easily see when CSI is approaching an extreme high or low.

Can the line color for all four lines be gray please? Preferably, the 1 Std Dev lines will be a lighter gray and the 2 Std Dev lines will be a medium gray (so there’s a little bit of contrast between the two pairs).

Have both activated will result in an output similar to the below:



Graph #2

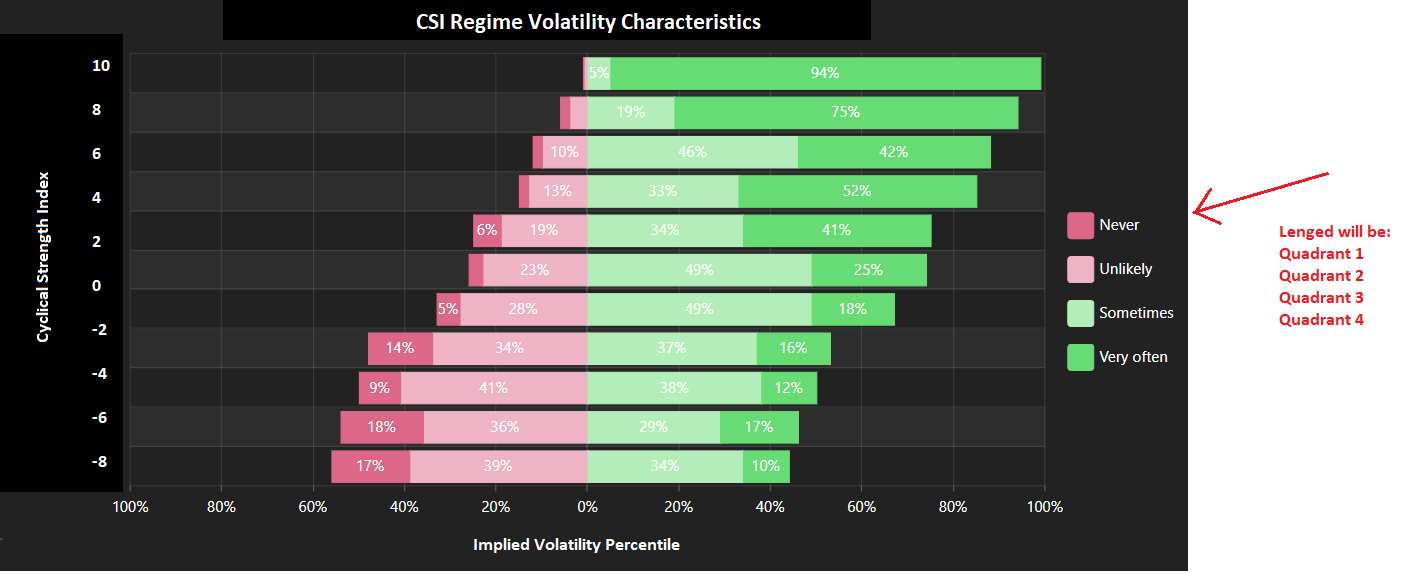
I would like to produce a similar graph as the following example:

<https://www.amcharts.com/demos/divergent-stacked-bars/?theme=dark>

This one will be a bit custom… so I will do my best to describe the desired output below.

I have included a CSV file to be used for the dataset. Do you mind creating a db table in our python anywhere postgres database to house this data? Unlike the CSI data, this table will not require periodic updating, so its ok if it only get uploaded a single time.

I have included a screenshot below where I made some edits to the chart in MS Paint. I added a title and axis labels.



Essentially, the chart’s purpose is to compare CSI and the volatility values in the CSV file. Although all the volatility values are between 0 -100, the chart’s x-axis actually has 0 in the middle and range from 100 to 100 on either side. To take advantage of this, we can divide up CSI into “good” and “bad” quadrants. Good quadrants are ones where the market tends to outperform, and bad quadrants are where the market tends to underperform. Quadrants 1 and 4 will be our good quadrants… while quadrants 2 and 3 will be our bad quadrants. Therefore, the volatility values in the CSV for quadrants 1 and 4 will be graphed on the right in green and the volatility values for quadrants 2 and 3 will be graphed on the left in red.

I have tried to point this out a bit more in the screenshot below:

