PART-TIME DATA SCIENCE - DATA VISUALIZATION

Data Visualization V...

Overhauling Matplotlib Defaults

Report a content mistake

Learning Objectives:

- Explore and set a "context" for visuals
- Adjust the relative font size

Using seaborn's `sns.set_context` to simplify resizing text

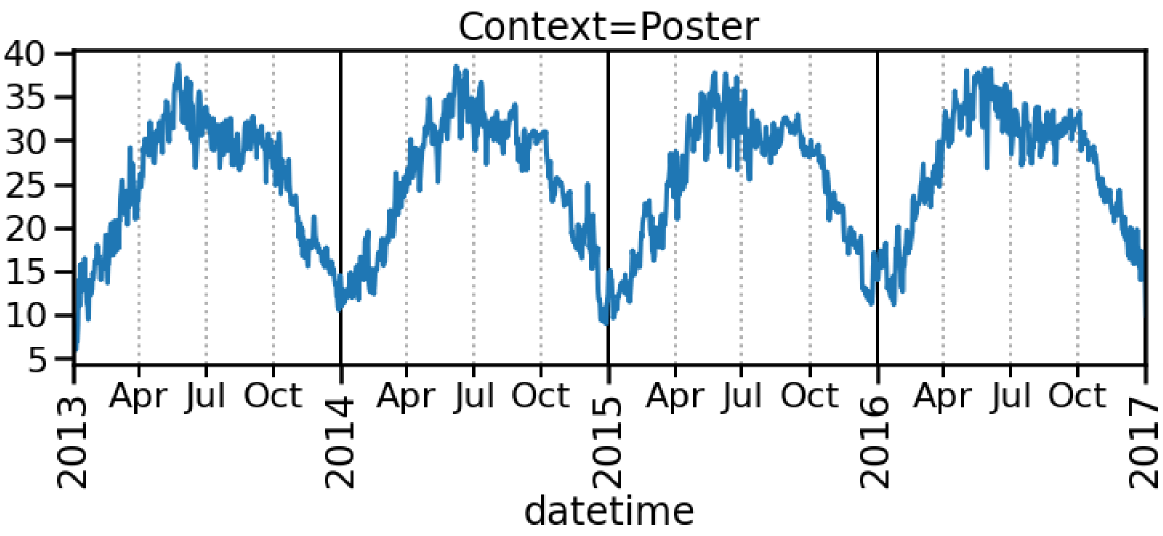
- Overall, we have a nice visualization thus far. One major consideration for our final figure is the overall text size and readability.
- Instead of manually setting the font size and parameters for every piece of text on our figure (major/minor tick labels, xlabel/ylabel, title, etc.):
    - we can use seaborn's `sns.set_context` function!
- According to the documentation for the set_context function, it will:
    - "*Set the parameters that control the scaling of plot elements. This affects things like the size of the labels, lines, and other elements of the plot, but not the overall style.*"
- The function accepts the following parameters:
    - "context":
        - The scenario where your figure will viewed.
        - Options are:
            - "notebook"
            - "talk"
            - "paper"
            - "poster"
    - "font_scale":
        - a float that represents how much more to scale text
        - 1=default size, 2= double size, 0.5 = half size, etc.

Temporarily using a context with `sns.plotting_context`

- If you only want to *temporarily* change the context settings, we can use sns.plotting_context (documentation).
    - We use the word "with" to temporarily apply the settings, just like when we temporarily opened local files.
- Example:
- We are going to first explore some options using the temporary sns.plotting_context. Once we have decided on our final format, we will use sns.set_context.

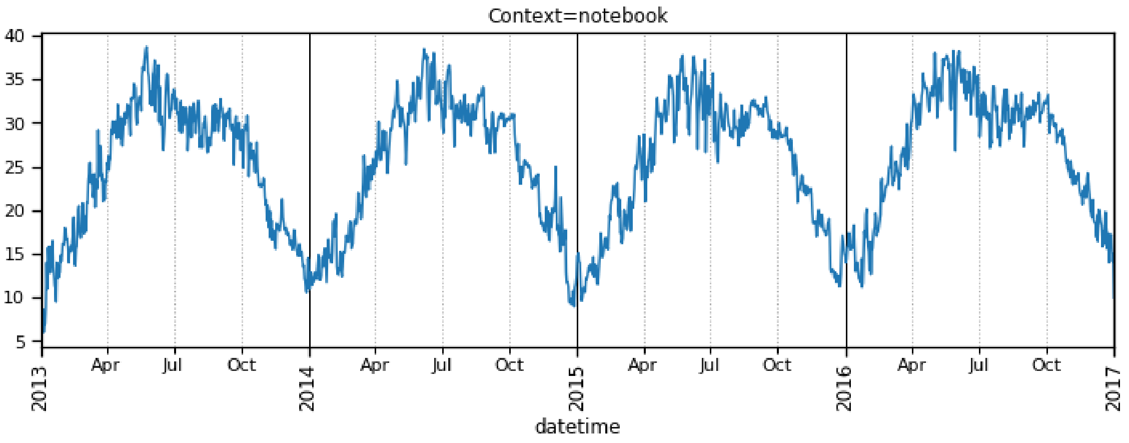
Note that this is using the custom function (plot_dates) we created in an earlier lesson!

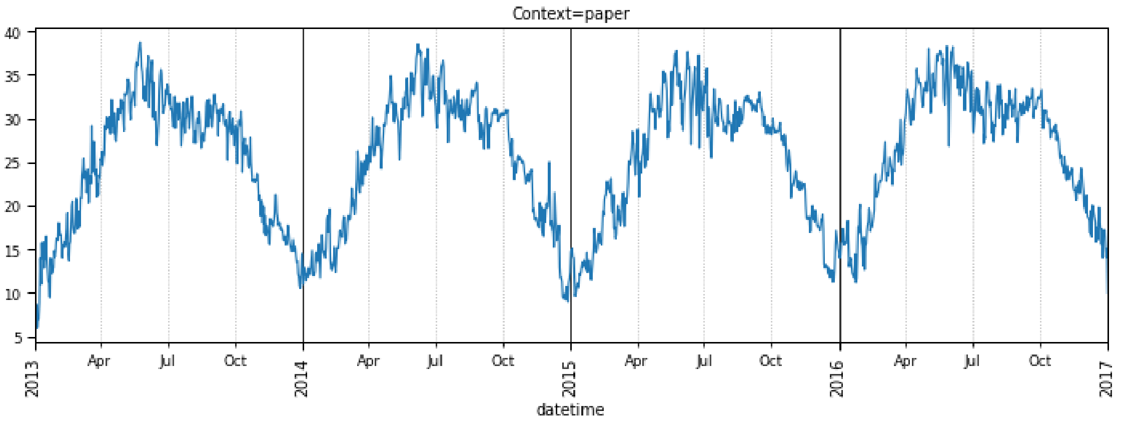
```
import seaborn as sns
with sns.plotting_context('poster'):
    fig,ax =
plot_dates(ts,xtick_fontsize=None,xtick_fontweight=None,title="Context=Poster")
```

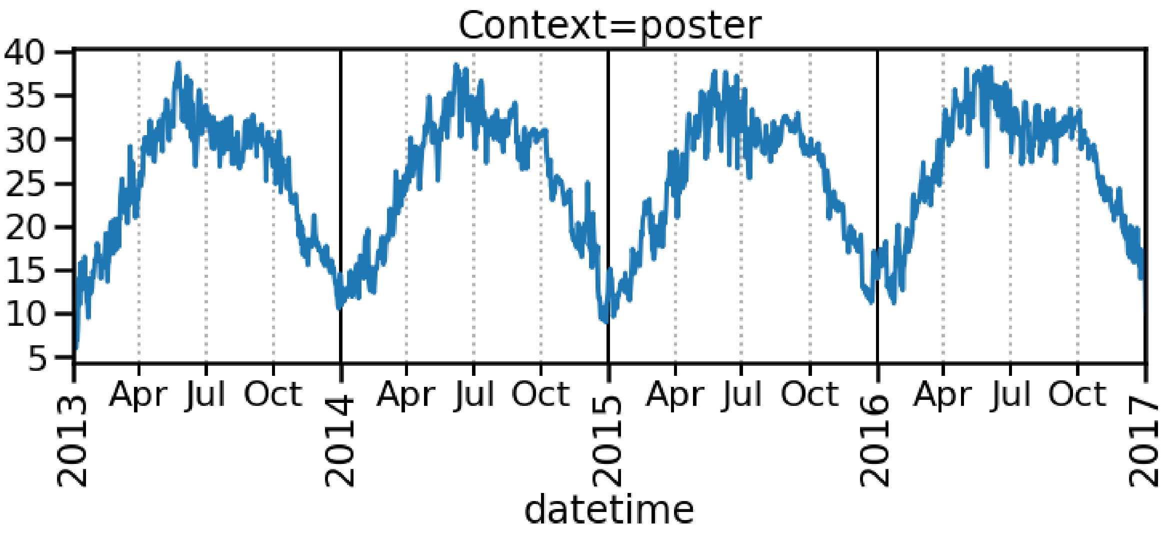


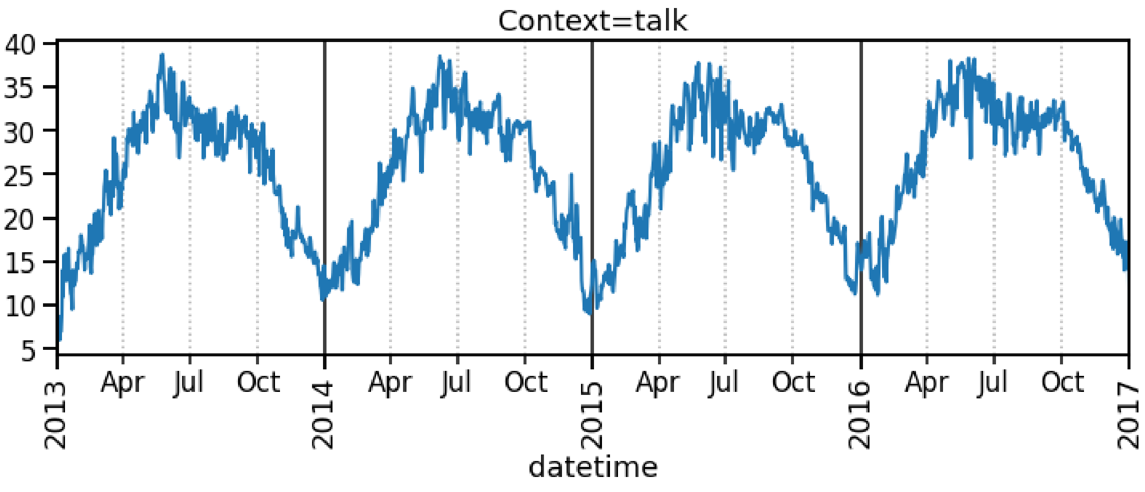
- Let's explore all of the context options at once. We will include the context in the title for reference as we view the outputs.

```
## temporarily changing context
for context in ['notebook','paper','poster','talk']:
    with sns.plotting_context(context):
        fig,ax = plot_dates(ts,xtick_fontsize=None,xtick_fontweight=None,
                            title=f"Context={context}")
```



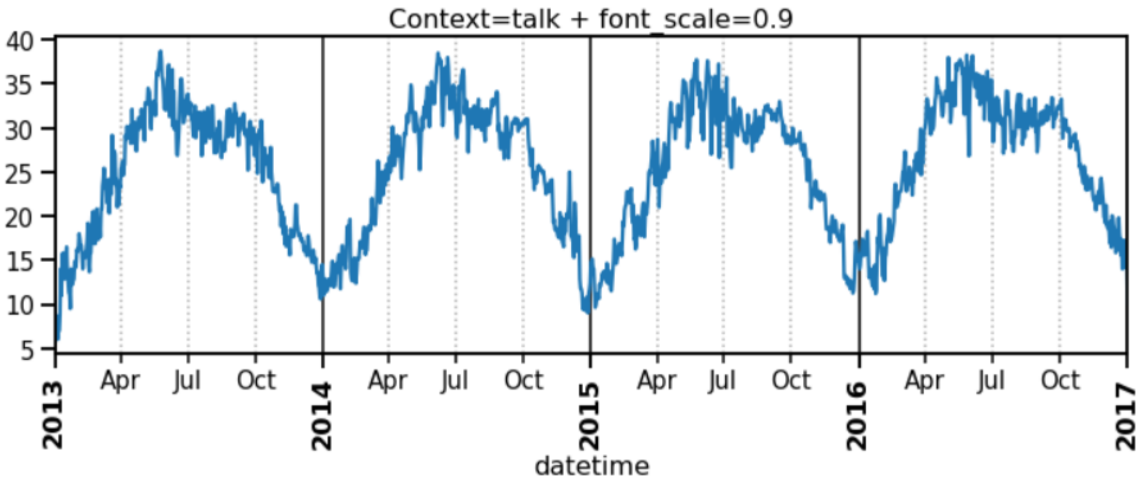






- As always, there are so many options for changing the style of your visuals! In this case, let's try using "talk" but also scaling the font a little smaller.

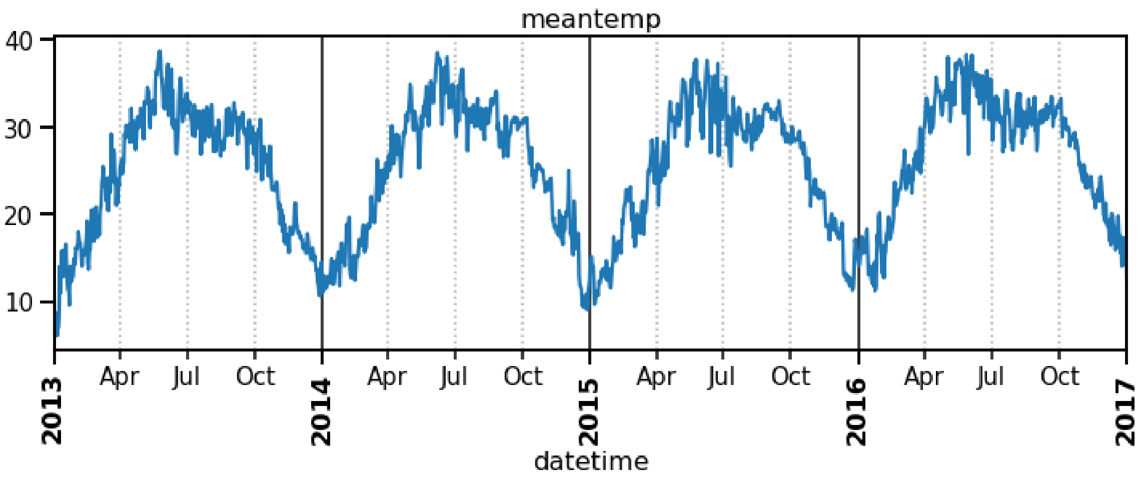
```
with sns.plotting_context("talk", font_scale=0.9):
    fig,ax = plot_dates(ts,xtick_fontsize=None,
                        xtick_fontweight="bold",
                        title=f"Context=talk + font_scale=0.9")
```



Ok, let's assume we are happy with the style of the graph above. Now we will make the settings permanent by using sns.set_context

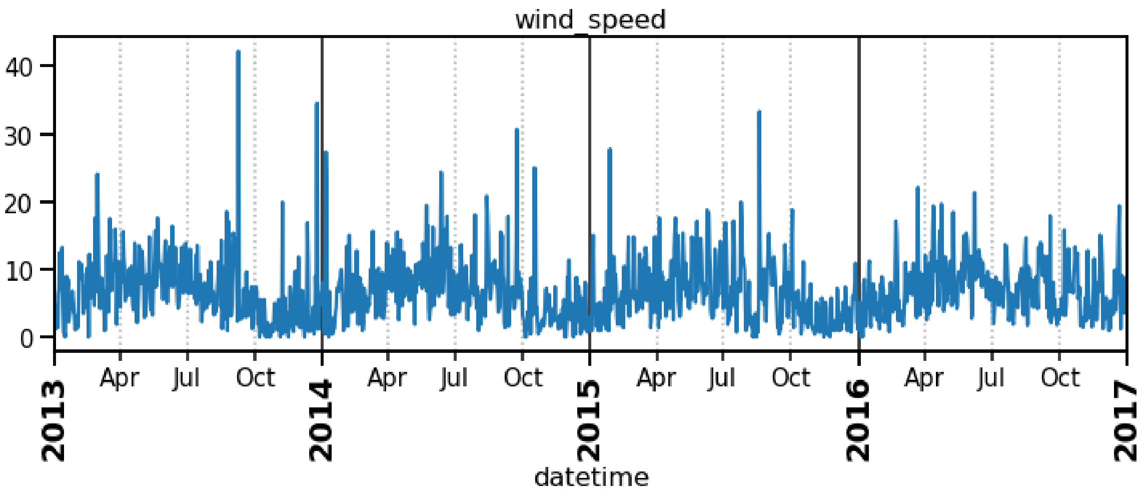
Using sns.set_context

```
## making the settings permanent for the notebook:
sns.set_context("talk", font_scale=0.9)
fig,ax = plot_dates(ts,xtick_fontsize=None,
                    xtick_fontweight="bold")
```



Now we can easily apply the same settings to the next graph in our notebook! Let's take a look at wind_speed instead of meantemp.

```
##  Visualize wind_speed instead of meantemp
ts = df['wind_speed'].copy()
plot_dates(ts);
```



- Notice how jagged our line for our time series is. The reason for this is because we seem to have values for every single DAY. But we are visualizing 4 full years of daily data!
- We can change the frequency of our time series and calculate aggregate measure (like the mean, sum, etc).
- In the next lesson we will learn how to resample time series with Pandas.

Summary

This lesson introduced Seaborn's context feature. We first temporarily explored the contexts

Previous

Next