Plotly Notes

- How do I install plotly?
- **Answer:** Go to the plotly page on the python package index for install instructions or see the plotly documentation. In this tutorial we have used the anaconda package environment to install plotly.

- What version of plotly is used for this tutorial?
- **Answer: 5.5**

- How do I make plots with plotly?
- **Answer:** Plotly plots can be made with plotly express, plotly graph objects, figure factories or dictionaries.

- What is plotly express?
- **Answer:** Plotly express is the easy-to-use, high-level interface to plotly, which operates on a variety of types of data and produces easy-to-style figures. It is recommended to start with plotly express however if more options are needed you can update the plotly express figure or start from scratch using a graph object. Figures made with plotly express can be customized in all the same ways as figures made with graph objects (e.g.-update and add methods), as well as with plotly express specific function arguments. The functions in plotly express, are all built on top of graph objects, and all return instances of plotly.graph_objects.Figure.

- What are plotly graph objects?
- **Answer:** Graph objects are the low-level building blocks of figures which you can use instead of plotly express for greater control.

- What are figure factories?
- **Answer:** Figure factories are dedicated functions for creating very specific types of plots. Note that some figure factories have been deprecated.

- What is the difference between plotly express and graph objects?
- **Answer:** Plotly express is to seaborn (high-level) as plotly graph objects is to matplotlib (lower-level). Most of the time, the same plots can be created with plotly express or graph objects. Usually, you can create plots with plotly express with much less code than graph objects however graph objects may provide greater control. Not all plots can be created with plotly express at this time and may require graph objects or figure factories.

- What is a trace?
- **Answer:** Think of a trace as a plot or part of a plot. Each trace has one of more than 40 possible types (e.g. scatter, bar, pie, surface, choropleth, etc.) and represents a set of related graphical marks in a figure.

- What is the difference between add_trace and append_trace?
- **Answer:** New traces can be added to a graph object figure using the add_trace() method. This method accepts a graph object trace (an instance of go.Scatter, go.Bar, etc.) and adds it to the figure. This allows you to start with an empty figure, and add traces to it sequentially. The append_trace() method does the same thing, although it does not return the figure.

- How do I add to or update a figure/plot?
- **Answer:** You can use update methods (e.g.-update_traces, update_layout, update_xaxes, update_yaxes, etc.) and/or add methods (e.g.-add_trace, add_annotation, add_shape, add_vline, add_hline, add_bar, add_scatter). You can use these methods to update or add to the figure/plot after a figure/plot is initially created.

- What is update_traces?
- **Answer:** You can set options common to all traces with the update_traces() method (examples - opacity, marker_color, marker_line_width, marker_size, marker_symbol, etc.).

- What is update_layout?
- **Answer:** You can use the update_layout() method to update the layout of a figure/plot (e.g.- update title, update legend, update figure/plot size, update style template).

- What is update_xaxes, update_yaxes?
- **Answer:** You can use the update_xaxes(), update_yaxes() methods to update the plot x and y axes (e.g.-title_text, tickangle).

- What kinds of data can be used with plotly?
- **Answer:** The kinds of data that can be used with plotly are long form, wide form, and mixed form (e.g.-dataframes, arrays, lists, and dictionaries).

- What is chart studio?
- **Answer:** Chart Studio provides a web-service for hosting graphs. Public hosting is free, for private hosting, there are paid plans. Graphs are saved inside your online Chart Studio account.

- Do plotly maps require and API key?
- **Answer:** Some plotly maps require an API key. The examples in this tutorial do not use an API key.

- What is the difference between Figure and FigureWidget?
- **Answer:** Plotly figures can be displayed in ipywidgets contexts using plotly.graph_objects.FigureWidget objects. FigureWidget is a figure graph object (just like plotly.graph_objects.Figure), so you can add traces to it and update it just like a regular Figure. But FigureWidget is also an ipywidgets object, which means that you can display it alongside other ipywidgets to build user interfaces right in the notebook. It is important to note that FigureWidget does not use the renderers framework, so you should not use the show() figure method or the plotly.io.show function on FigureWidget objects.

- How do I change a renderer?
- **Answer:** To change a renderer use: import plotly.io as pio, pio.renderers.default = 'browser' or fig.show(renderer='png'). Default renderer: 'notebook_connected'. Available renderers: 'plotly_mimetype', 'jupyterlab', 'nteract', 'vscode', 'notebook', 'notebook_connected', 'kaggle', 'azure', 'colab', 'cocalc', 'databricks', 'json', 'png', 'jpeg', 'jpg', 'svg', 'pdf', 'browser', 'firefox', 'chrome', 'chromium', 'iframe', 'iframe_connected', 'sphinx_gallery', 'sphinx_gallery_png'

- How do I speed things up if I am working with a really big data set?
- **Answer:** To speed things up for large data sets use the orjson package, datashader, and scattergl() for scatter plots.

- Can I use plotly with Google Colab, Spyder, Jupyter Notebooks, JupyterLab, Visual Studio Code?
- **Answer:** Yes. Some of these options may require some extra setup. I was able to create plots using plotly in Google Colab, Jupyter Notebooks, JupyterLab without extra setup however this will vary. Spyder - Install kaleido (recommended) or orca (legacy) in package environment. To open in the default browser use - import plotly.io as pio, pio.renderers.default='browser'. To show plots inline in Spyder plot section - use pio.renderers.default='svg'. Visual Studio Code - I was able to create plotly plots and display them in the browser using visual studio code and the anaconda plotly package environment. Remember to use fig.show() to display the plot. fig.show() is not always necessary but it appears to be required in this case.

- Can I use a pandas backend?
- **Answer:** Yes, plotly express can be used as a pandas .plot() backend. Use import pandas as pd, pd.options.plotting.backend = 'plotly', fig = df.plot(). You can also use the cufflinks package.