PharmaSUG China 2022 - Paper DV-126

Interactive Data Visualization with Python Plotly

Wei (Tony) Zhang, Pfizer (China) Research and Development Co.,Ltd. China

ABSTRACT

Plotly is an open-source module of Python which is used for data visualization and supports various graphs, such as line charts, scatter plots, bar charts, histograms, area charts, etc. There are several libraries available in Python, such as Matplotlib, Seaborn. for data visualization. However, they render only the static images of the plots. Due to this, many crucial information get lost in visualization.
Plotly allows us to have advantage of being more interactive. The graphs are stored in JSON data format and can be read by other scripting languages. Plotly can be used to create online and offline charts. In this article, we present how to plot a basic chart with Plotly and customize the chart for interactive data visualization. Before starting, a basic knowledge of Python such as string, tuple, list, dictionary is helpful to understand the options in the Plotly.

INTRODUCTION

There are many plotting libraries in Python The most popular ones are: Matplotlib, Seaborn, Plotly, and Bokeh. Matplotlib is the oldest Python plotting library, and it's still the most popular. It was created in 2003 as part of the SciPy Stack, an open source scientific computing library similar to Matlab. Seaborn is an abstraction layer on top of Matplotlib; it gives you a really neat interface to make a wide range of useful plot types very easily. Plotly is a plotting ecosystem that includes a Python plotting library. It has three different interfaces: an object-oriented interface; an imperative interface that allows you to specify your plot using JSON-like data structures; a high-level interface similar to Seaborn called Plotly Express. Plotly plots are designed to be embedded in web apps. At its core, Plotly is actually a JavaScript library! It uses D3 and stack.gl to draw the plots. Bokeh (pronounced "BOE-kay") specializes in building interactive plots, so this standard example doesn't show it off to its best. Like Plotly, Bokeh's plots are designed to be embedded in web apps; it outputs its plots as HTML files.

PLOTLY OPEN SOURCE GRAPHING LIBRARY FOR PYTHON

Plotly's Python graphing library makes interactive, publication-quality graphs. Examples of how to make line plots, scatter plots, area charts, bar charts, error bars, box plots, histograms, heatmaps, subplots, multiple-axes, polar charts, and bubble charts. Plotly.py is free and open source and you can view the source, report issues or contribute on GitHub.

INSTALLATION

plotly may be installed using pip:

$ pip install plotly

or conda:

$ conda install -c plotly plotly

This package contains everything you need to write figures to standalone HTML files.

You can try the following Python code to generate a plot. A HTML graphic file is generated with open status.

```
import plotly.express as px
fig = px.bar(x=["a", "b", "c"], y=[1, 3, 2])
fig.write_html('first_figure.html', auto_open=True)
```

PLOTLY PACKAGE

The main modules in Plotly are:

- plotly.plotly
- plotly.graph.objects
- plotly.express

plotly.plotly acts as the interface between the local machine and Plotly. It contains functions that require a response from Plotly's server.

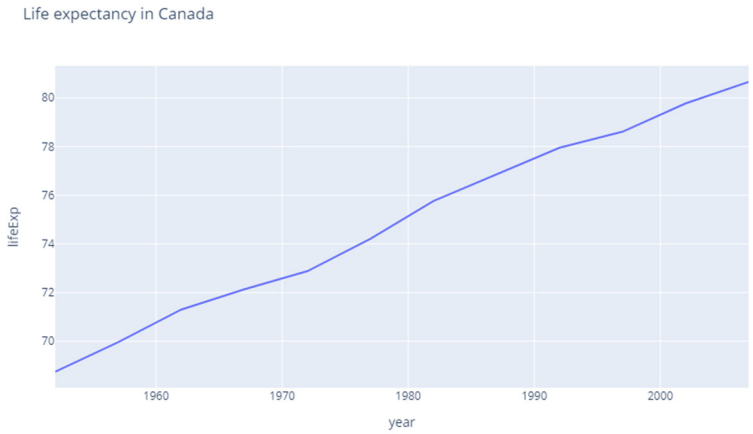
plotly.graph_objects module contains the objects (Figure, layout, data, and the definition of the plots like scatter plot, line chart) that are responsible for creating the plots.

plotly.express module can create the entire Figure at once. It uses the graph_objects internally and returns the graph_objects.Figure instance.

LINE CHART

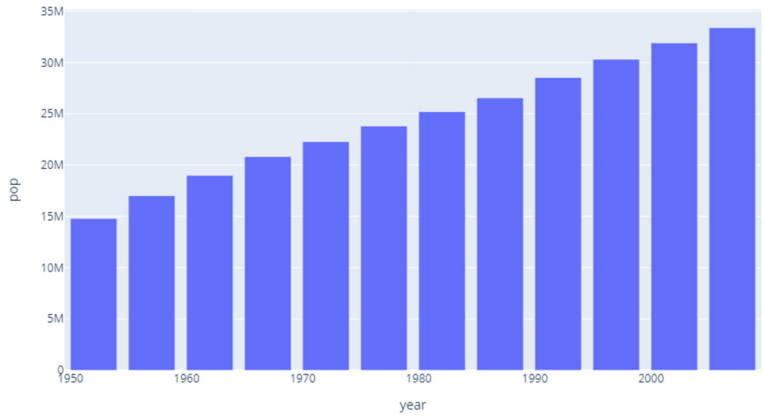
```
import plotly.express as px


df = px.data.gapminder().query("country=='Canada'")
fig = px.line(df, x="year", y="lifeExp", title='Life expectancy in Canada')
fig.show()
```



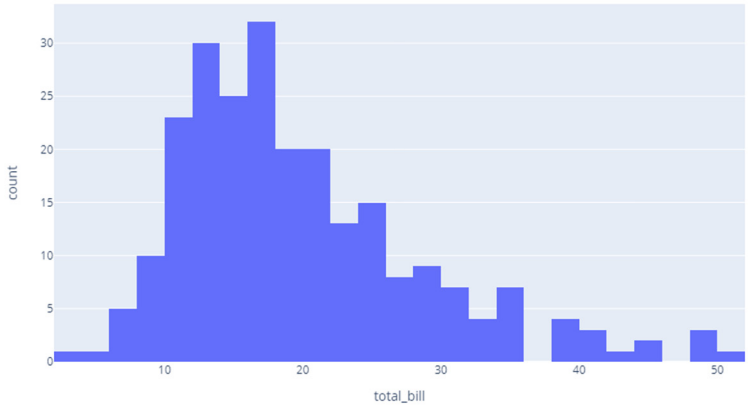
BAR CHART

```
import plotly.express as px
data_canada = px.data.gapminder().query("country == 'Canada'")
fig = px.bar(data_canada, x='year', y='pop')
fig.show()
```



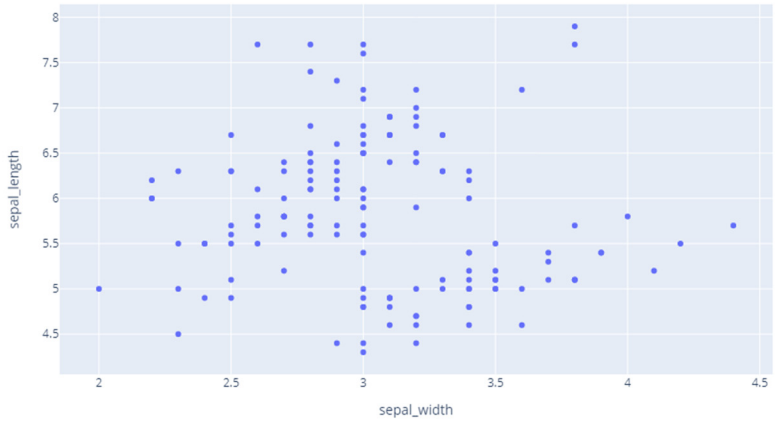
HISTOGRAMS

```
import plotly.express as px
df = px.data.tips()
fig = px.histogram(df, x="total_bill")
fig.show()
```



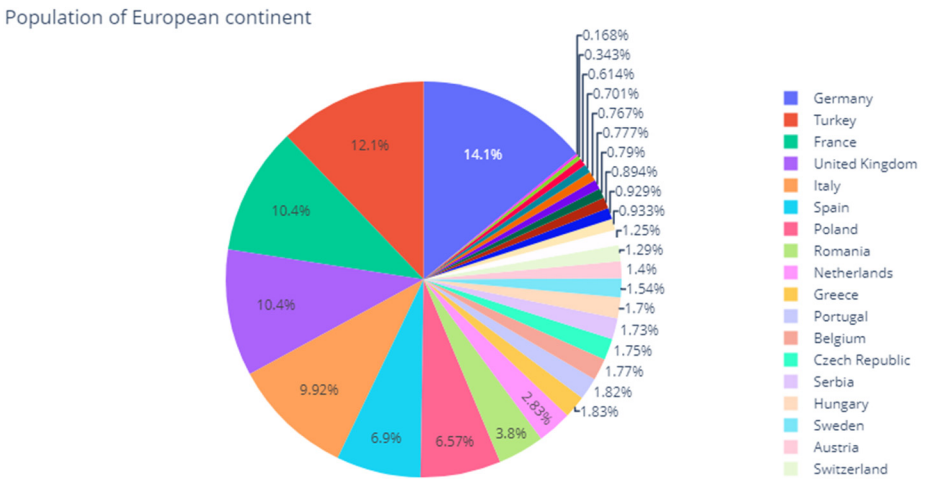
SCATTER PLOTS

```
import plotly.express as px
df = px.data.iris() # iris is a pandas DataFrame
fig = px.scatter(df, x="sepal_width", y="sepal_length")
fig.show()
```



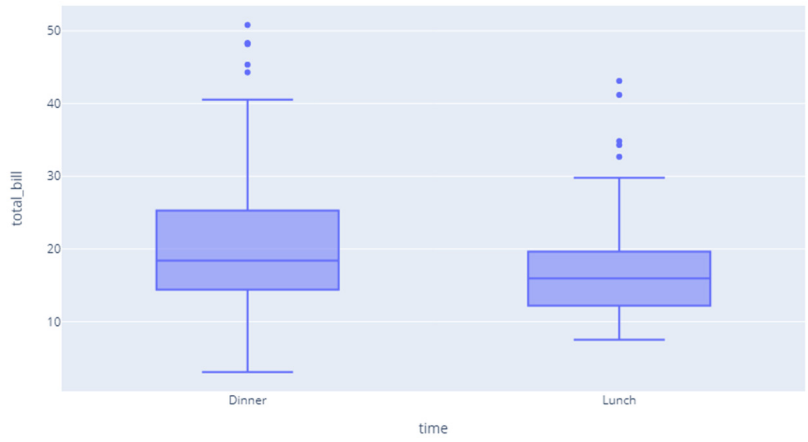
PIE CHARTS

```
import plotly.express as px
df = px.data.gapminder().query("year == 2007").query("continent == 'Europe'")
df.loc[df['pop'] < 2.e6, 'country'] = 'Other countries' # Represent only large countries
fig = px.pie(df, values='pop', names='country', title='Population of European continent')
fig.show()
```



BOX PLOTS

```
import plotly.express as px
df = px.data.tips()
fig = px.box(df, x="time", y="total_bill")
fig.show()
```



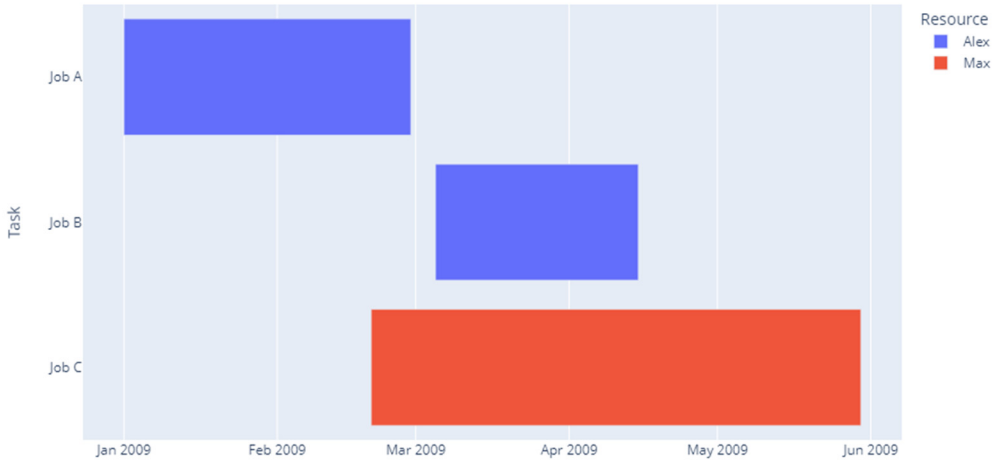
GANTT CHARTS

```
import plotly.express as px
```

```
import pandas as pd

df = pd.DataFrame([
    dict(Task="Job A", Start='2009-01-01', Finish='2009-02-28', Resource="Alex"),
    dict(Task="Job B", Start='2009-03-05', Finish='2009-04-15', Resource="Alex"),
    dict(Task="Job C", Start='2009-02-20', Finish='2009-05-30', Resource="Max")
])

fig = px.timeline(df, x_start="Start", x_end="Finish", y="Task", color="Resource")
fig.update_yaxes(autorange="reversed")
fig.show()
```

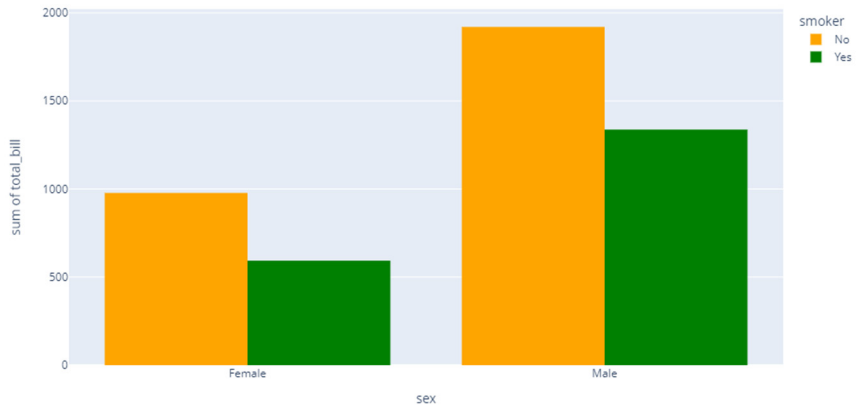


CUSTOMIZE YOUR PLOT

- **Colors**

  This is the example to change the color in the color_discrete_suquence.

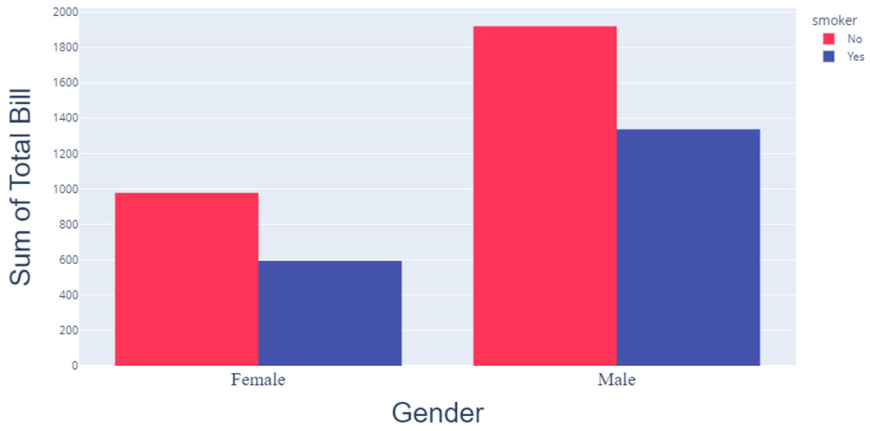
```
import plotly.express as px
df = px.data.tips()
fig = px.histogram(df, x="sex", y="total_bill",
            color='smoker', barmode='group',
            color_discrete_sequence=['orange', 'green'])
fig.show()
```



- **Fonts**

  This is the example to change the axis fonts and title name in fig.update_xaxes.

  fig.update_xaxes(**tickfont=dict(family='Times New Roman', size=20), title_font=dict(size=30, family='Arial'), title_text='Gender', dtick=200**)

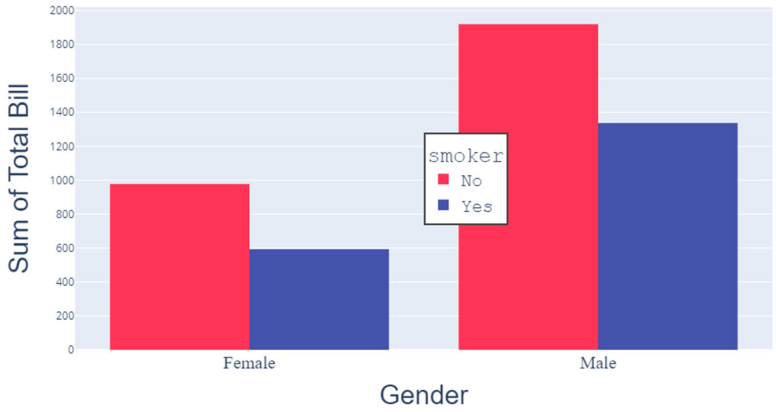


- **Legend**

  This is the example to change the legend location in fig.update_layout

  fig.update_layout(showlegend=True, **legend_font=dict(size=20, family='Courier New'), legend_borderwidth=2, legend_x=0.5, legend_y=0.5**)

  Note: in the fig.update_layout(legend_x=n, legend_y=n), n betweens or equals to -2 to 3 for the location.



CONCLUSION

According to the TIOBE index, which measures the popularity of programming languages, Python is the most popular programming language in the world. There are many reasons for the ubiquity of Python, including its ease of use, simple syntax, thriving community and versatility.

This paper is just an introduction of Python plotly which can be used to generate the plots in Python for data visualization. Plotly produces interactive graphs, can be embedded on websites, and provides a wide variety of complex plotting options. The graphs and plots are robust and a wide variety of people can use them. The visuals are of high quality and easy to read and interpret.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Name: Wei (Tony) Zhang
Enterprise: Pfizer (China) Research and Development Co.,Ltd. China
E-mail: wei.zhang18@pfizer.com