Plotly

Library for visualizations

Plotly

- The plotly.express module (usually imported as px) contains functions that can create entire figures at once, and is referred to as Plotly Express or PX.
- Plotly Express is a built-in part of the plotly library, and is the recommended starting point for creating most common figures.
- Plotly Express provides more than 30 functions for creating different types of figures.

Plotly dataset

- There are different built-in dataset available with plotly
- plotly.data.iris()- Each row represents a flower wit['sepal_length', 'sepal_width', 'petal_length', 'petal_width', 'species', 'species_id’]
- Can convert that into pandas dataframe and view.
- medals_long-This dataset represents the medal table for Olympic Short Track Speed Skating for the top three nations as of 2020.
- medals_wide-This dataset represents the medal table for Olympic Short Track Speed Skating for the top three nations as of 2020.

Contd..

- Gapminder()-Dataset describing life expentency depending on factors like fertility, GDP etc

Scatter plots

- A **scatter plot** is a diagram where each value is represented by the dot graph.
- Scatter plot needs arrays for the same length, one for the value of x-axis and other value for the y-axis.
- Syntax: plotly.express.scatter(data_frame=None, x=None, y=None, color=None, symbol=None, size=None, hover_name=None, hover_data=None, custom_data=None, text=None, facet_row=None, facet_col=None, facet_col_wrap=0, error_x=None, error_x_minus=None, error_y=None, error_y_minus=None, animation_frame=None, animation_group=None, category_orders={}, labels={}, orientation=None, color_discrete_sequence=None, color_discrete_map={}, color_continuous_scale=None, range_color=None, color_continuous_midpoint=None, symbol_sequence=None, symbol_map={}, opacity=None, size_max=None, marginal_x=None, marginal_y=None, trendline=None, trendline_color_override=None, log_x=False, log_y=False, range_x=None, range_y=None, render_mode='auto', title=None, template=None, width=None, height=None)

Line chart in plotly

- Line plot in Plotly is much accessible and illustrious annexation to plotly which manage a variety of types of data and assemble easy-to-style statistic.
- With px.line each data position is represented as a vertex transformed(which location is given by the x and y columns) of a polyline mark in 2D space.

Line chart

- plotly.express.line(data_frame=None, x=None, y=None, line_group=None, color=None, line_dash=None, hover_name=None, hover_data=None, custom_data=None, text=None, facet_row=None, facet_col=None, facet_col_wrap=0, error_x=None, error_x_minus=None, error_y=None, error_y_minus=None, animation_frame=None, animation_group=None, category_orders={}, labels={}, orientation=None, color_discrete_sequence=None, color_discrete_map={}, line_dash_sequence=None, line_dash_map={}, log_x=False, log_y=False, range_x=None, range_y=None, line_shape=None, render_mode='auto', title=None, template=None, width=None, height=None)

Filled Area Plots

- Filled area plot is the easy-to-use, high-level articulation to plotly which completes a variety of types of data and produces easy-to-style figures.
- Each filled area harmonizes with one value of the column given by the line_group parameter.
- Filled area charts are most commonly used to show trends, rather than convey specific values.

Filled area

- *plotly.express.area(data_frame=None, x=None, y=None, line_group=None, color=None, hover_name=None, hover_data=None, custom_data=None, text=None, facet_row=None, facet_col=None, facet_col_wrap=0, animation_frame=None, animation_group=None, category_orders={}, labels={}, color_discrete_sequence=None, color_discrete_map={}, orientation=None, groupnorm=None, log_x=False, log_y=False, range_x=None, range_y=None, line_shape=None, title=None, template=None, width=None, height=None)*

Bar chart

- In a bar chart the data categories are displayed on the vertical axis and the data values are displayed on the horizontal axis.
- Labels are easier to display and with a big data set they impel to work better in a narrow layout such as mobile view.

Bar chart

- plotly.express.bar(data_frame=None, x=None, y=None, color=None, facet_row=None, facet_col=None, facet_col_wrap=0, hover_name=None, hover_data=None, custom_data=None, text=None, error_x=None, error_x_minus=None, error_y=None, error_y_minus=None, animation_frame=None, animation_group=None, category_orders={}, labels={}, color_discrete_sequence=None, color_discrete_map={}, color_continuous_scale=None, range_color=None, color_continuous_midpoint=None, opacity=None, orientation=None, barmode='relative', log_x=False, log_y=False, range_x=None, range_y=None, title=None, template=None, width=None, height=None)

Funnel Chart

- Funnel charts are often used to represent data in different stages of a business process.
- It’s an important mechanism in Business Intelligence to identify potential problem areas of a process.

Gant charts

- A Gantt chart is a type of bar chart that illustrates a project schedule.
- The chart lists the tasks to be performed on the vertical axis, and time intervals on the horizontal axis.
- The width of the horizontal bars in the graph shows the duration of each activity.
- The px.timeline function by default sets the X-axis to be of type=date, so it can be configured like any time-series chart.

Deprecated Figure Factory

- Another package from plotly to create gant chart
- Prior to the introduction of plotly.express.timeline() in version 4.9, the recommended way to make Gantt charts was to use the now-deprecated create_gantt() figure factory.

Pie Charts in Python

- A pie chart is a circular statistical chart, which is divided into sectors to illustrate numerical proportion.

Sunburst Charts in Python

- Sunburst plots visualize hierarchical data spanning outwards radially from root to leaves.
- The root starts from the center and children are added to the outer rings.

Treemap Charts in Python

- Treemap charts visualize hierarchical data using nested rectangles.
- The input data format is the same as for Sunburst Charts and Icicle Charts: the hierarchy is defined by labels (names for px.treemap) and parents attributes.
- B :The amount of padding (in px) along the bottom of the component.
- L :The amount of padding (in px) on the left side of the component.
- R:The amount of padding (in px) on the right side of the component.
- T:The amount of padding (in px) along the top of the component.

Icicle Charts in Python

- Icicle charts visualize hierarchical data using rectangular sectors that cascade from root to leaves in one of four directions: up, down, left, or right.

Histograms in Python

- histogram is representation of the distribution of numerical data, where the data are binned and the count for each bin is represented.
- histogram is an aggregated bar chart, with several possible aggregation functions (e.g. sum, average, count...) which can be used to visualize data on categorical and date axes as well as linear axes.

Box Plots in Python

- A box plot is a statistical representation of the distribution of a variable through its quartiles.
- The ends of the box represent the lower and upper quartiles, while the median (second quartile) is marked by a line inside the box.

Violin Plots in Python

- A violin plot is a statistical representation of numerical data. It is similar to a box plot, with the addition of a rotated kernel density plot on each side.

Strip Charts in Python

- Strip charts are like 1-dimensional jittered scatter plots.

Empirical Cumulative Distribution Plots

- Empirical cumulative distribution function plots are a way to visualize the distribution of a variable, and Plotly Express has a built-in function, px.ecdf() to generate such plots.
- 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.

2D Histograms in Python

- A 2D histogram, also known as a density heatmap, is the 2-dimensional generalization of a histogram which resembles a heatmap but is computed by grouping a set of points specified by their x and y coordinates into bins, and applying an aggregation function such as count or sum (if z is provided) to compute the color of the tile representing the bin.

3D Line Plots

- With px.line_3d each data position is represented as a vertex (which location is given by the x, y and z columns) of a polyline mark in 3D space.

Title maps

- **Mapbox maps** are tile-based maps.
- To draw a line on your map, you either can use px.line_mapbox() in Plotly Express.
- line_mapbox(data_frame=None, lat=None, lon=None, color=None, text=None, hover_name=None, hover_data=None, custom_data=None, line_group=None, animation_frame=None, animation_group=None, category_orders=None, labels=None, color_discrete_sequence=None, color_discrete_map=None, zoom=8, center=None, mapbox_style=None, title=None, template=None, width=None, height=None)

Outline maps

- line_geo(data_frame=None, lat=None, lon=None, locations=None, locationmode=None, geojson=None, featureidkey=None, color=None, line_dash=None, text=None, facet_row=None, facet_col=None, facet_col_wrap=0, facet_row_spacing=None, facet_col_spacing=None, hover_name=None, hover_data=None, custom_data=None, line_group=None, symbol=None, animation_frame=None, animation_group=None, category_orders=None, labels=None, color_discrete_sequence=None, color_discrete_map=None, line_dash_sequence=None, line_dash_map=None, symbol_sequence=None, symbol_map=None, markers=False, projection=None, scope=None, center=None, fitbounds=None, basemap_visible=None, title=None, template=None, width=None, height=None)

Polar charts

- These charts use a polar coordinate system, where its x-axis looks like a circle with the origin point as a center and each point is determined by distance from a fixed point and angle from a fixed direction.
- Polar charts are also known as radar charts, web charts, spider charts, and star charts, and many others.
- The radial and angular coordinates names are given as r and theta arguments.

Polar charts

- A polar chart represents data along radial and angular axes. With Plotly Express, it is possible to represent polar data as scatter markers with px.scatter_polar, and as lines with px.line_polar.
- *plotly.express.scatter_polar(data_frame=None, r=None, theta=None, color=None, symbol=None, size=None, hover_name=None, hover_data=None, title=None, template=None, width=None, height=None)*

Polar charts

- line_polar(data_frame=None, r=None, theta=None, color=None, line_dash=None, hover_name=None, hover_data=None, custom_data=None, line_group=None, text=None, symbol=None, animation_frame=None, animation_group=None, category_orders=None, labels=None, color_discrete_sequence=None, color_discrete_map=None, line_dash_sequence=None, line_dash_map=None, symbol_sequence=None, symbol_map=None, markers=False, direction='clockwise', start_angle=90, line_close=False, line_shape=None, render_mode='auto', range_r=None, range_theta=None, log_r=False, title=None, template=None, width=None, height=None)