plotly | Graphing Libraries (https://plotly.com/)(/graphing-libraries/)

utm_campaign=studio_cloud_launch&utm_content=sidebar)

Python (/python) > *Statistical Charts (/python/statistical-charts)* > *Marginal Distribution Plots*

Suggest an edit to this (https://github.com/plotly/plotly.py/edit/doc-page   prod/doc/python/marginal-plots.md)

Marginal Distribution Plots in Python

How to add marginal distribution plots.

> Plotly Studio: Transform any dataset into an interactive data application in minutes with AI. Sign up for early access now. (https://plotly.com/studio/?utm_medium=graphing_libraries&utm_campaign=studio_early_access&utm_content=sidebar)

Overview

Marginal distribution plots are small subplots above or to the right of a main plot, which show the distribution of data along only one dimension. Marginal distribution plot capabilities are built into various Plotly Express functions such as scatter and histogram. Plotly Express (/python/plotly-express/) is the easy-to-use, high-level interface to Plotly, which operates on a variety of types of data (/python/px-arguments/) and produces easy-to-style figures (/python/styling-plotly-express/).
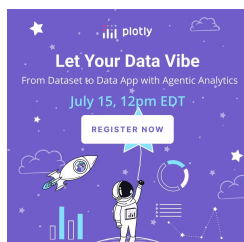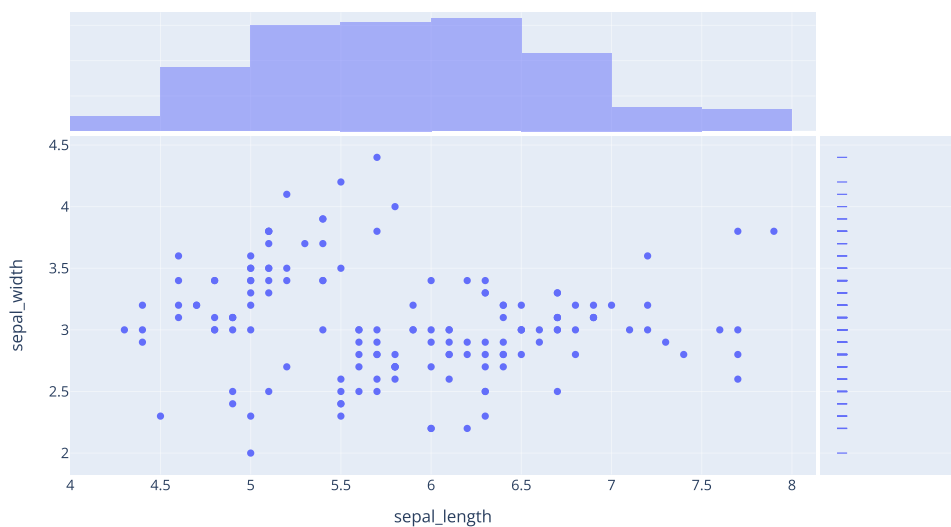
Scatter Plot Marginals

The marginal_x and marginal_y arguments accept one of "histogram", "rug", "box", or "violin" (see also how to create histograms (/python/histograms/), box plots (/python/box-plots/) and violin plots (/python/violin-plots/) as the main figure).
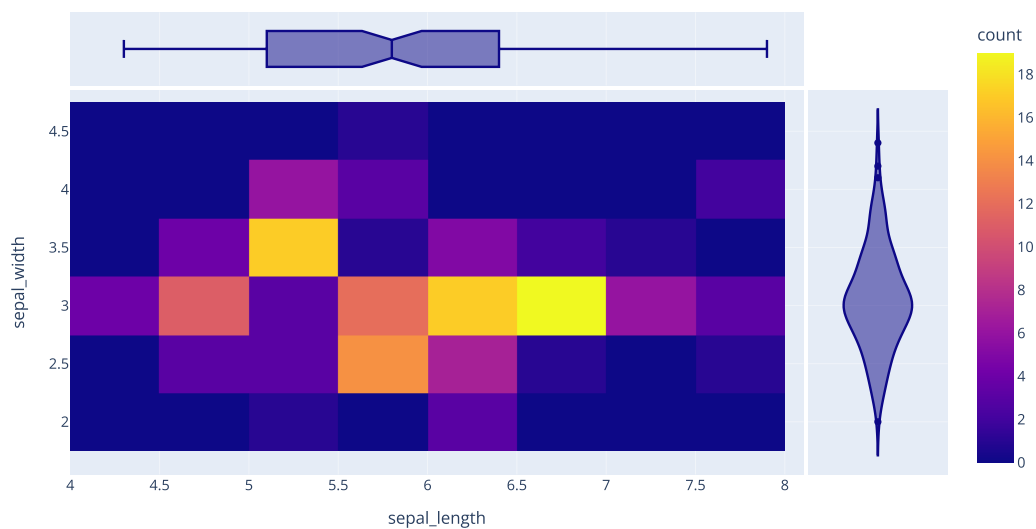
Marginal plots are linked to the main plot: try zooming or panning on the main plot.

Marginal plots also support hover, including per-point hover as with the rug-plot on the right: try hovering over the points on the right marginal plot.

```python
import plotly.express as px
df = px.data.iris()
fig = px.scatter(df, x="sepal_length", y="sepal_width", marginal_x="histogram", marginal_y="rug")
fig.show()
```

```
import plotly.express as px
df = px.data.iris()
fig = px.density_heatmap(df, x="sepal_length", y="sepal_width", marginal_x="box", marginal_y="violin")
fig.show()
```
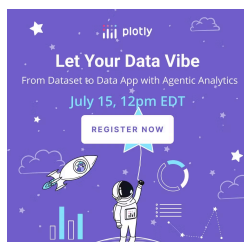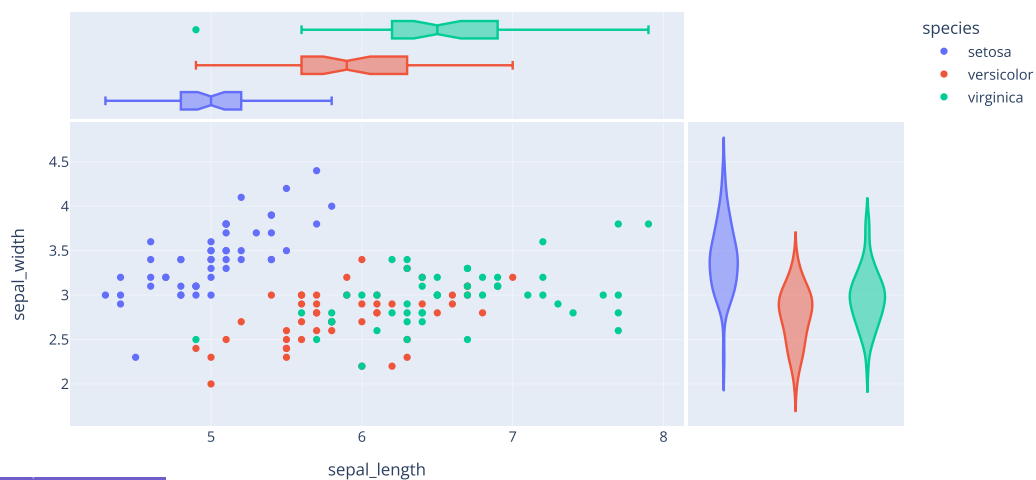

Marginal Plots and Color

Marginal plots respect the color argument as well, and are linked to the respective legend elements. Try clicking on the legend items.

```
import plotly.express as px
df = px.data.iris()
fig = px.scatter(df, x="sepal_length", y="sepal_width", color="species",
                 marginal_x="box", marginal_y="violin",
                  title="Click on the legend items!")
fig.show()
```
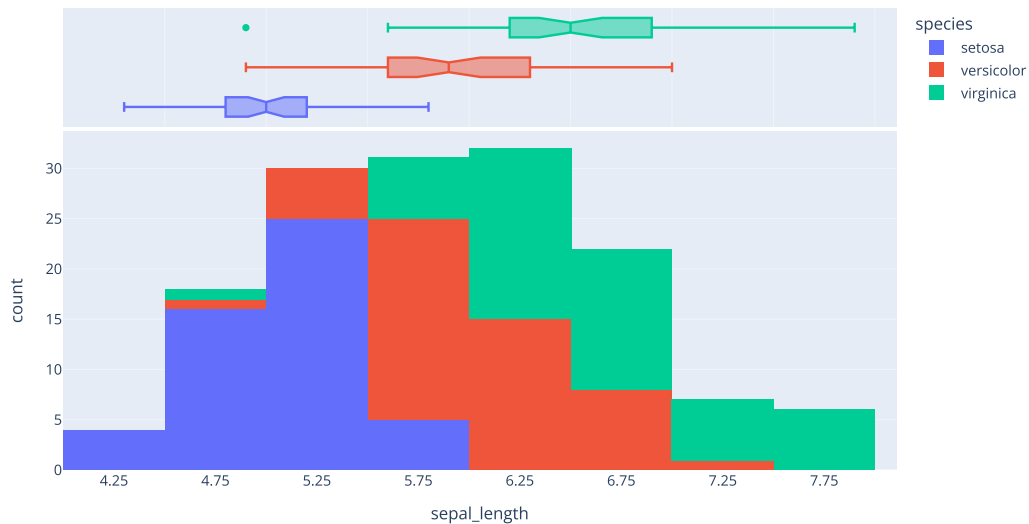
Click on the legend items!

Marginal Plots on Histograms

Histograms (/python/histograms/) are often used to show the distribution of a variable, and they also support marginal plots in Plotly Express, with the marginal argument:

```python
import plotly.express as px
df = px.data.iris()
fig = px.histogram(df, x="sepal_length", color="species", marginal="box")
fig.show()
```
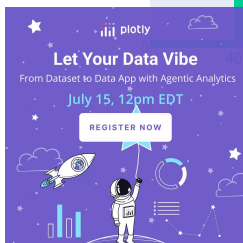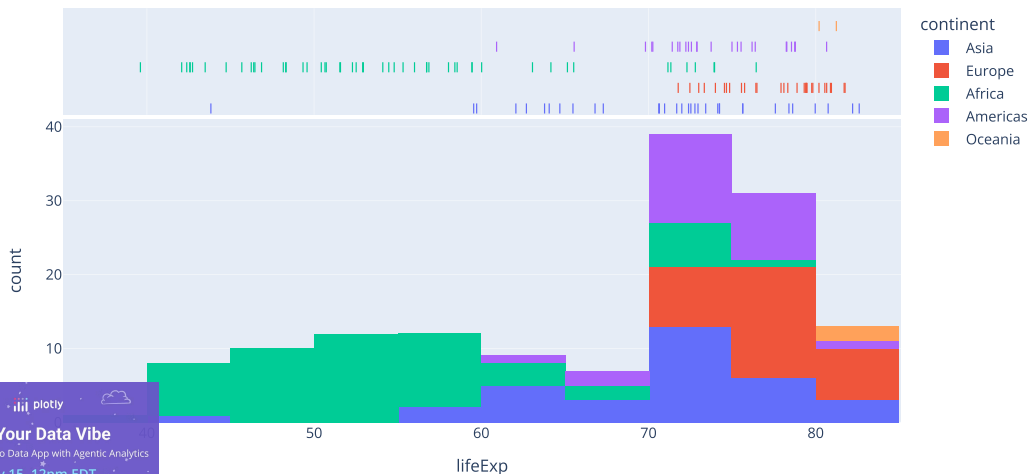

Try hovering over the rug plot points to identify individual country values in the histogram below:

```python
import plotly.express as px
df = px.data.gapminder().query("year == 2007")
fig = px.histogram(df, x="lifeExp", color="continent", marginal="rug", hover_name="country",
                   title="Hover over the rug plot!")
fig.show()
```
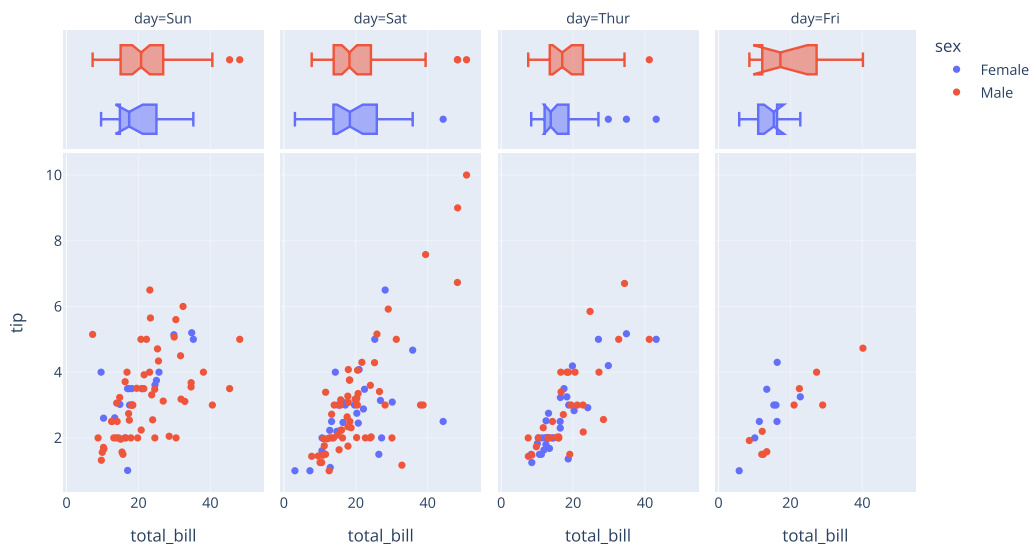
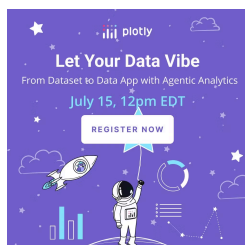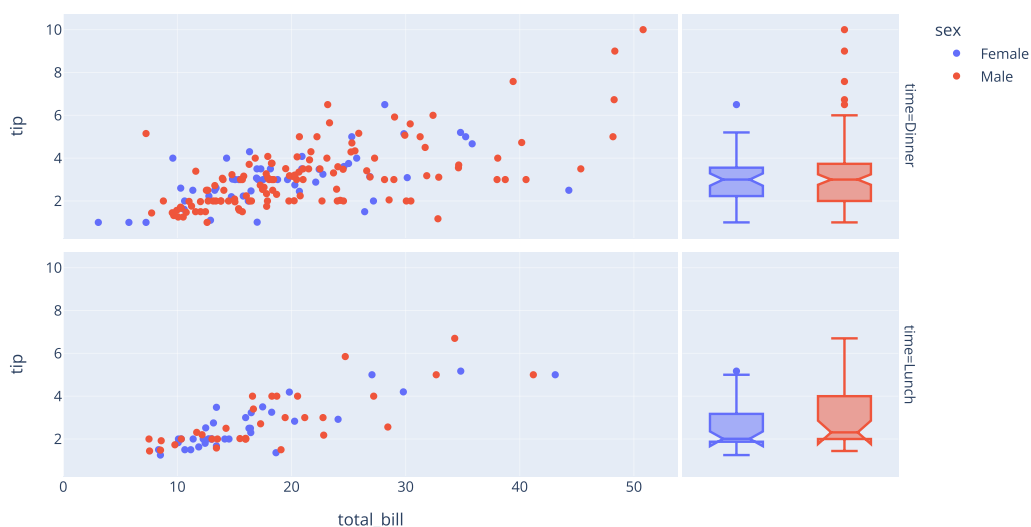## Marginal Plots and Facets

Marginal plots can be used in conjunction with Plotly Express facets (/python/facet-plots/) so long as they go along different directions:
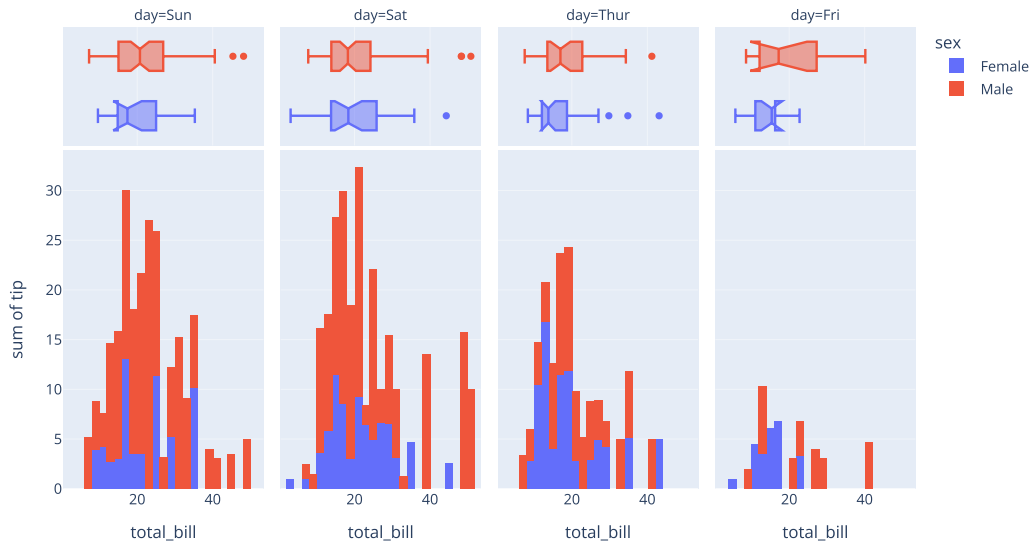
```python
import plotly.express as px
df = px.data.tips()
fig = px.scatter(df, x="total_bill", y="tip", color="sex", facet_col="day",
                 marginal_x="box")
fig.show()
```



```python
import plotly.express as px
df = px.data.tips()
fig = px.scatter(df, x="total_bill", y="tip", color="sex", facet_row="time",
                 marginal_y="box")
fig.show()
```

```
import plotly.express as px
df = px.data.tips()
fig = px.histogram(df, x="total_bill", y="tip", color="sex", facet_col="day",
                   marginal="box")
fig.show()
```


What About Dash?

Dash (https://dash.plot.ly/) is an open-source framework for building analytical applications, with no Javascript required, and it is tightly integrated with the Plotly graphing library.

Learn about how to install Dash at https://dash.plot.ly/installation (https://dash.plot.ly/installation).

Everywhere in this page that you see fig.show(), you can display the same figure in a Dash application by passing it to the figure argument of the Graph component (https://dash.plot.ly/dash-core-components/graph) from the built-in dash_core_components package like this:

```
import plotly.graph_objects as go # or plotly.express as px
fig = go.Figure() # or any Plotly Express function e.g. px.bar(...)
# fig.add_trace( ... )
# fig.update_layout( ... )

from dash import Dash, dcc, html

app = Dash()
app.layout = html.Div([
    dcc.Graph(figure=fig)
])

app.run(debug=True, use_reloader=False)  # Turn off reloader if inside Jupyter
```


...tutorial?utm_medium=graphing_libraries&utm_content=python_footer)

JOIN OUR MAILING LIST

Sign up to stay in the loop with all things Plotly — from Dash Club
to product updates, webinars, and more!

SUBSCRIBE
(HTTPS://GO.PLOT.LY/SUBSCRIPTION)

Products

Dash (https://plotly.com/dash/)

Consulting and Training
(https://plotly.com/consulting-and-oem/)

Pricing

Enterprise Pricing (https://plotly.com/get-pricing/)

About Us

Careers (https://plotly.com/careers)

Resources (https://plotly.com/resources/)

Blog (https://medium.com/@plotlygraphs)

Support

Community Support (https://community.plot.ly/)

Documentation (https://plotly.com/graphing-libraries)

Terms of Service (https://community.plotly.com/tos)   Privacy Policy (https://plotly.com/privacy/)