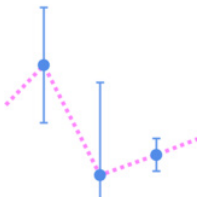
Python (/python) > Statistical Charts (/python/statistical-charts)

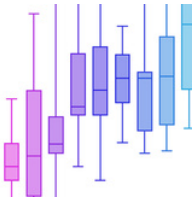
Plotly Python Open Source Graphing Library Statistical Charts

Plotly's Python graphing library makes interactive, publication-quality graphs online. Examples of how to make statistical charts such as box plots, histograms, and distribution plots.



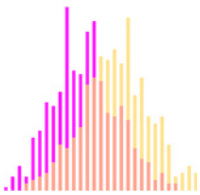
(/python/error-bars/)

Error Bars



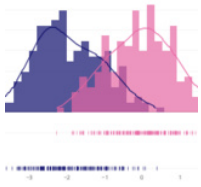
(/python/box-plots/)

Box Plots



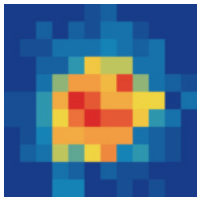
(/python/histograms/)

Histograms



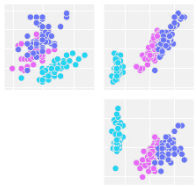
(/python/distplot/)

Distplots



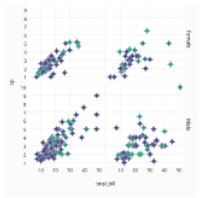
(/python/2D-Histogram/)

2D Histograms



(/python/splom/)

Scatterplot Matrix

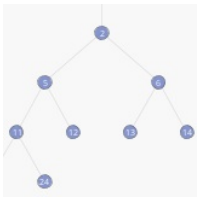


(/python/facet-plots/)

Facet and Trellis Plots

(/python/parallel-categories-diagram/)

Parallel Categories Diagram

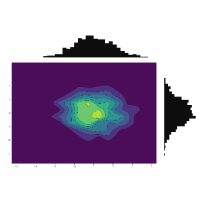


(/python/tree-plots/)

Tree-plots

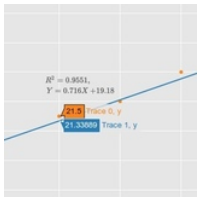
(/python/violin/)

Violin Plots



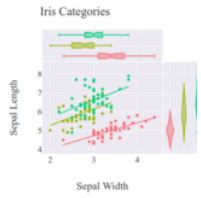
(/python/2d-histogram-contour/)

2D Histogram Contour



(/python/linear-fits/)

Linear and Non-Linear Trendlines

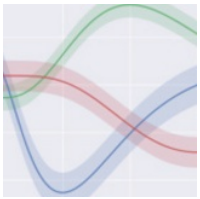


(/python/marginal-plots/)

Marginal Distribution Plots

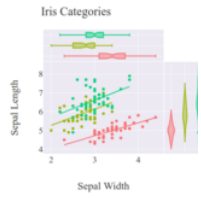
(/python/strip-charts/)

Strip Charts



(/python/continuous-error-bars/)

Continuous Error Bands



(/python/ecdf-plots/)

Empirical Cumulative Distribution Plots

Statistical charts in Dash

Dash (https://plotly.com/dash/) is the best way to build analytical apps in Python using Plotly figures. To run the app below, run `pip install dash`, click "Download" to get the code and run `python app.py`.

Get started with the official Dash docs (https://dash.plotly.com/installation) and learn how to effortlessly style (https://plotly.com/dash/design-kit/) & deploy (https://plotly.com/dash/app-manager/) apps like this with Dash Enterprise (https://plotly.com/dash/).

```python
from dash import Dash, dcc, html, Input, Output
import plotly.express as px
import numpy as np

app = Dash(__name__)


app.layout = html.Div([
    html.H4('Interactive normal distribution'),
    dcc.Graph(id="graph"),
    html.P("Mean:"),
    dcc.Slider(id="mean", min=-3, max=3, value=0,
               marks={-3: '-3', 3: '3'}),
    html.P("Standard Deviation:"),
    dcc.Slider(id="std", min=1, max=3, value=1,
               marks={1: '1', 3: '3'}),
])


@app.callback(
    Output("graph", "figure"),
    Input("mean", "value"),
    Input("std", "value"))
def display_color(mean, std):
    data = np.random.normal(mean, std, size=500) # replace with your own data source
```

DOWNLOAD

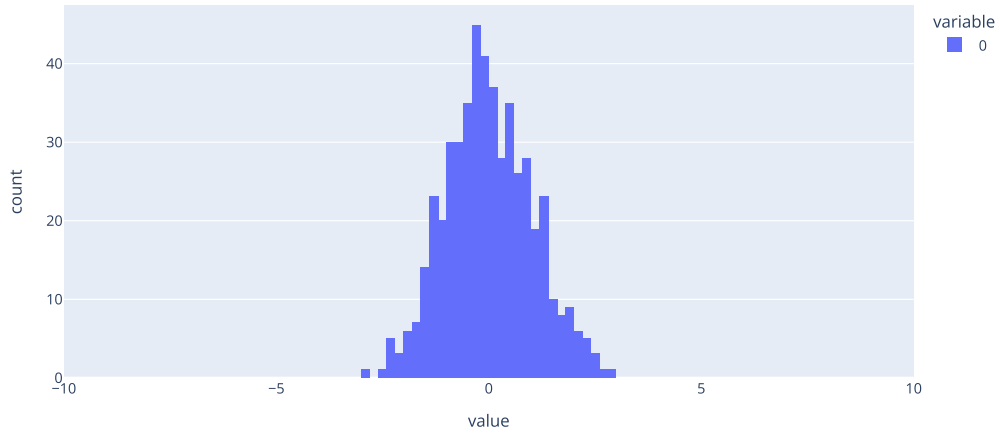
rted)
:)
e/index/)
:e)
ial)
lotly.py)

l-charts/)
charts/)
harts/)

bio/)

ons)

Interactive normal distribution



Mean:

-3 3

Standard Deviation:

1 3

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)