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

utm_campaign=studio_cloud_launch&utm_content=sidebar)

Python (/python) > (/python/) > Peak Finding   ◆ Suggest an edit to this page(https://github.com/plotly/plotly.py/edit/doc-prod/doc/python/peak-finding.md)

Peak Finding in Python

Learn how to find peaks and valleys on datasets in Python

> 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)

Imports

The tutorial below imports Pandas (https://pandas.pydata.org/docs/user_guide/10min.html), and SciPy (https://www.scipy.org/).

```
import pandas as pd
from scipy.signal import find_peaks
```

Import Data

To start detecting peaks, we will import some data on milk production by month:

```
import plotly.graph_objects as go
import pandas as pd

milk_data = pd.read_csv('https://raw.githubusercontent.com/plotly/datasets/master/monthly-milk-production-pounds.csv')
time_series = milk_data['Monthly milk production (pounds per cow)']

fig = go.Figure(data=go.Scatter(
    y = time_series,
    mode = 'lines'
))

fig.show()
```
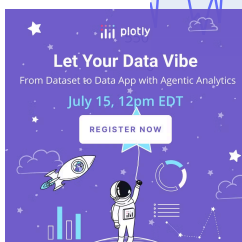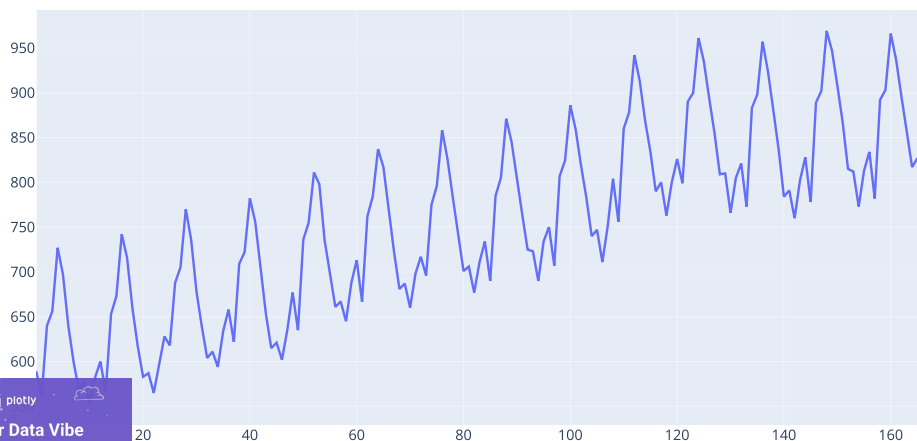
Peak Detection

We need to find the x-axis indices for the peaks in order to determine where the peaks are located.

```python
import plotly.graph_objects as go
import pandas as pd
from scipy.signal import find_peaks

milk_data = pd.read_csv('https://raw.githubusercontent.com/plotly/datasets/master/monthly-milk-production-pounds.csv')
time_series = milk_data['Monthly milk production (pounds per cow)']

indices = find_peaks(time_series)[0]

fig = go.Figure()
fig.add_trace(go.Scatter(
    y=time_series,
    mode='lines+markers',
    name='Original Plot'
))

fig.add_trace(go.Scatter(
    x=indices,
    y=[time_series[j] for j in indices],
    mode='markers',
    marker=dict(
        size=8,
        color='red',
        symbol='cross'
    ),
    name='Detected Peaks'
))

fig.show()
```
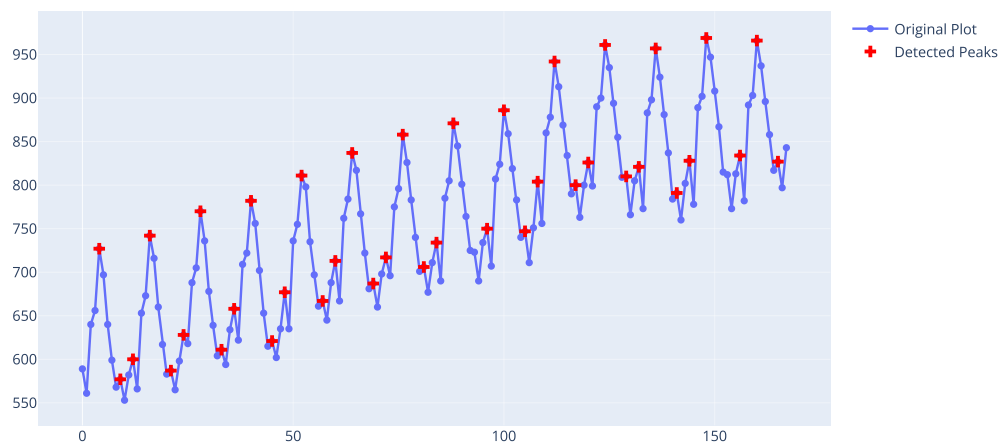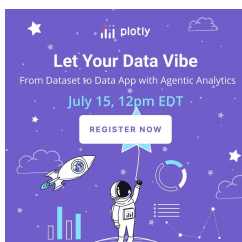

Only Highest Peaks

We can attempt to set our threshold so that we identify as many of the *highest peaks* that we can.

```python
import plotly.graph_objects as go
import numpy as np
import pandas as pd
from scipy.signal import find_peaks

milk_data = pd.read_csv('https://raw.githubusercontent.com/plotly/datasets/master/monthly-milk-production-pounds.csv')
time_series = milk_data['Monthly milk production (pounds per cow)']

indices = find_peaks(time_series, threshold=20)[0]

fig = go.Figure()
fig.add_trace(go.Scatter(
    y=time_series,
    mode='lines+markers',
    name='Original Plot'
))

fig.add_trace(go.Scatter(
    x=indices,
    y=[time_series[j] for j in indices],
    mode='markers',
    marker=dict(
        size=8,
        color='red',
        symbol='cross'
    ),
    name='Detected Peaks'
))

fig.show()
```
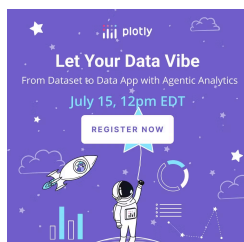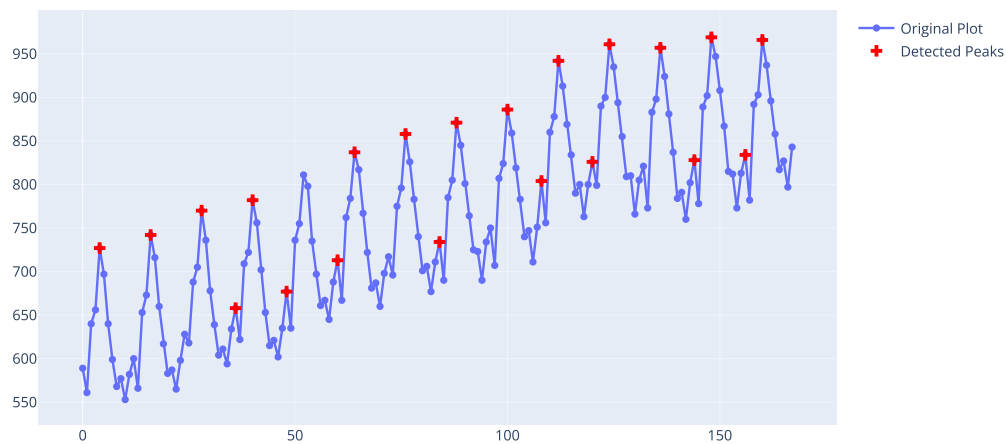
What About Dash?

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:

```python
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
```


(https://dash.plotly.com/tutorial?utm_medium=graphing_libraries&utm_content=python_footer)