

You have **1** free member-only story left this month. [Upgrade for unlimited access.](#)

PYTHON FOR DATA VISUALIZATION

# How to Build Dashboards with Python

Visualizing Data in Dashboards through Python Libraries



Roberto Salazar

Follow



Oct 29 · 3 min read ★





Image by [Franki Chamaki](#) available at [Unsplash](#)

## Data Visualization through Dashboards

Visualizing data and results clearly and concisely represents a critical step in any data analytics project. At the end of it, it is all about how well you can communicate your findings to others to facilitate and improve the decision-making process. Critical business decisions require supporting data to ensure the best alternatives are chosen, and it is your responsibility as a data analyst to provide such.

**Your data needs to tell a story and its representation should be easy to read, understand and interpret for a non-technical audience.** While you might be a technical expert with strong programming, analytical, statistical and mathematical background, you can't assume that everyone else in the room will have the same technical knowledge as you. For that reason, you should avoid losing the audience concentration and engagement at all costs due to highly-technical or complex concepts.

Python, one of today's most popular programming languages around the world, has multiple data visualization open-source libraries that allow its users to build interactive dashboards with simple lines of code. While it does not follow a drag-and-drop approach as other data visualization programs, its degree of complexity is still simple enough for new programmers to learn quickly.

. . .

For the following example, let's build a dashboard for analyzing Tesla's stock from January 2020 up to October 2021. Let's take a look at the Python code!

```
1 # Import required libraries
2 import yfinance as yf
```

```

2 import yfinance as yf
3 import pandas as pd
4 import hvplot.pandas
5 import panel as pn
6 pn.extension('plotly')
7 import plotly.express as px
8 import plotly.graph_objects as go
9
10 # Get stock data
11 TSLA_data = yf.Ticker("TSLA").history(period='1d', start='2020-1-1', end='2021-10-29')
12
13 # Visualize data frame top rows
14 TSLA_data.head()

```

Dashboards #1.py hosted with ❤ by GitHub

[view raw](#)

	Open	High	Low	Close	Volume	Dividends	Stock Splits
Date							
2020-01-02	84.900002	86.139999	84.342003	86.052002	47660500	0	0.0
2020-01-03	88.099998	90.800003	87.384003	88.601997	88892500	0	0.0
2020-01-06	88.094002	90.311996	88.000000	90.307999	50665000	0	0.0
2020-01-07	92.279999	94.325996	90.671997	93.811996	89410500	0	0.0
2020-01-08	94.739998	99.697998	93.646004	98.428001	155721500	0	0.0

```

1 # Select 'Close' column only
2 TSLA = TSLA_data[["Close"]].copy()
3
4 # Add a daily return column
5 TSLA["Daily_Return"] = TSLA["Close"].pct_change()

```

```

6
7 # Add a simple moving average column of window of 10 days
8 TSLA["SMA10"] = TSLA["Close"].rolling(window=10).mean()
9
10 # Add a simple moving average column of window of 50 days
11 TSLA["SMA50"] = TSLA["Close"].rolling(window=50).mean()
12
13 # Add a simple moving average column of window of 100 days
14 TSLA["SMA100"] = TSLA["Close"].rolling(window=100).mean()
15
16 # Visualize data frame top rows
17 TSLA.head()

```

Dashboards #2.py hosted with ❤ by GitHub

[view raw](#)

	Close	Daily_Return	SMA10	SMA50	SMA100
Date					
2020-01-02	86.052002	NaN	NaN	NaN	NaN
2020-01-03	88.601997	0.029633	NaN	NaN	NaN
2020-01-06	90.307999	0.019255	NaN	NaN	NaN
2020-01-07	93.811996	0.038801	NaN	NaN	NaN
2020-01-08	98.428001	0.049205	NaN	NaN	NaN

After we have gathered all the required data, we can proceed to build the plots that will be displayed on the dashboard.

```

1 # Create an hvplot table for the stock data
2 TSLA_df_table = TSLA.hvplot.table(width=1000)

```

```

3
4 # Create an hvplot line plot for the closing price
5 TSLA_closing_price_plot = TSLA.drop(columns="Daily_Return").hvplot.line(title="TSLA Closing Price
6
7 # Create an hvplot line plot for the daily return
8 TSLA_daily_retrun_plot = TSLA["Daily_Return"].hvplot.line(title="TSLA Daily Returns", ylabel="Da
9
10 # Create an hvplot bar plot for the daily volume
11 TSLA_trading_volume_barplot = TSLA_data["Volume"].hvplot.bar(title="TSLA Trading Volume", ylabel
12
13 # Create a candlestick plot for the stock data
14 TSLA_candlestick = go.Figure(data=[go.Candlestick(x=TSLA_data.index,
15             open=TSLA_data['Open'],
16             high=TSLA_data['High'],
17             low=TSLA_data['Low'],
18             close=TSLA_data['Close'])])
19 TSLA_candlestick.update_layout(height=800, width=1000, title='TSLA Candlestick')

```

Dashboards #3.py hosted with ❤ by GitHub

[view raw](#)

Now that we have built all the necessary plots, let's configure the dashboard, as shown below:

. . .

**Note:** You might need to run the following command on your terminal window to get the dashboard visualized:

```
jupyter labextension install @pyviz/jupyterlab_pyviz
```

**Source:** <https://panel.holoviz.org/>

. . .

```
1  # Create a title for the dashboard
2  title = pn.pane.Markdown(
3      """
4  # Tesla Stock Analysis Dashboard
5  """,
6  width=1000,
7  )
8
9  # Create a welcome message for the dashboard
10 welcome = pn.pane.Markdown(
11     """
12     This dashboard presents a visual analysis of Telsa stock (ticker: "TSLA") from January 2020 to 0
13     You can navigate through the tabs below and interact with the plots to explore more details about
14     ""
15 )
16
17 # Define the ditinct tabs for the dashboard with their corresponding plots
18 tabs = pn.Tabs(
19     ("Historical Data", pn.Column(TSLA_df_table)),
20     ("Candlestick Plot", pn.Column(TSLA_candlestick)),
21     ("Closing Price and Volume", pn.Column(TSLA_closing_price_plot, TSLA_trading_volume_barplot)),
22     ("Daily Returns", pn.Column(TSLA_daily_retrun_plot)),
```

```

23 )
24
25 # Build dashboard
26 dashboard = pn.Column(pn.Column(title, welcome), tabs, width=1000)
27
28 # Visualize dashboard
29 dashboard.servable()

```

Dashboards #4.py hosted with ❤ by GitHub

[view raw](#)

## Welcome Tab

### Tesla Stock Analysis Dashboard

This dashboard presents a visual analysis of Tesla stock (ticker: "TSLA") from January 2020 to October 2021. You can navigate through the tabs below and interact with the plots to explore more details about the Tesla stock.

Historical Data						
Candlestick Plot						
Closing Price and Volume						
Daily Returns						
#	Date	Close	Daily_Return	SMA10	SMA50	SMA100
0	2020-01-02 00:00:00	86.052002	NaN	NaN	NaN	NaN
1	2020-01-03 00:00:00	88.601997	0.029633	NaN	NaN	NaN
2	2020-01-06 00:00:00	90.307999	0.019255	NaN	NaN	NaN
3	2020-01-07 00:00:00	93.811996	0.038801	NaN	NaN	NaN
4	2020-01-08 00:00:00	98.428001	0.049205	NaN	NaN	NaN
5	2020-01-09 00:00:00	96.267998	-0.021945	NaN	NaN	NaN
6	2020-01-10 00:00:00	95.629997	-0.006627	NaN	NaN	NaN
7	2020-01-13 00:00:00	104.972	0.097689	NaN	NaN	NaN
8	2020-01-14 00:00:00	107.584	0.024883	NaN	NaN	NaN
9	2020-01-15 00:00:00	103.699997	-0.036102	96.535599	NaN	NaN
10	2020-01-16 00:00:00	102.697998	-0.009662	98.200198	NaN	NaN

## Candlestick Plot Tab



# Tesla Stock Analysis Dashboard

This dashboard presents a visual analysis of Tesla stock (ticker: "TSLA") from January 2020 to October 2021. You can navigate through the tabs below and interact with the plots to explore more details about the Tesla stock.

Historical Data **Candlestick Plot** Closing Price and Volume Daily Returns

TSLA Candlestick



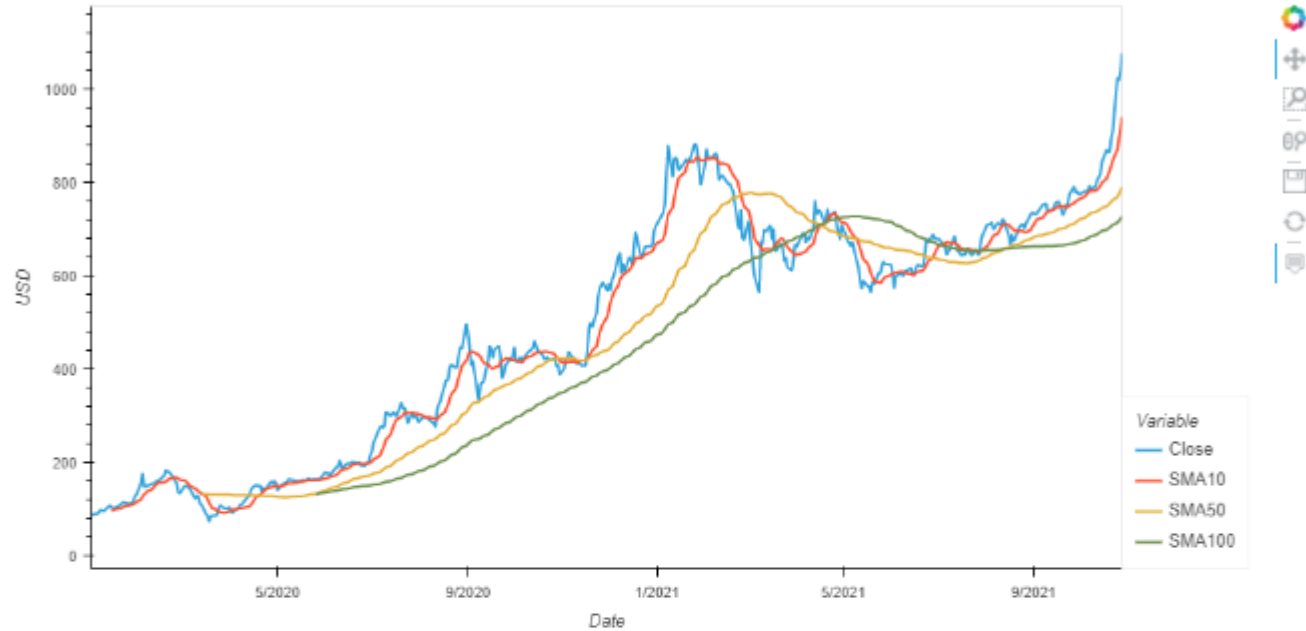
## Closing Price and Volume Tab

# Tesla Stock Analysis Dashboard

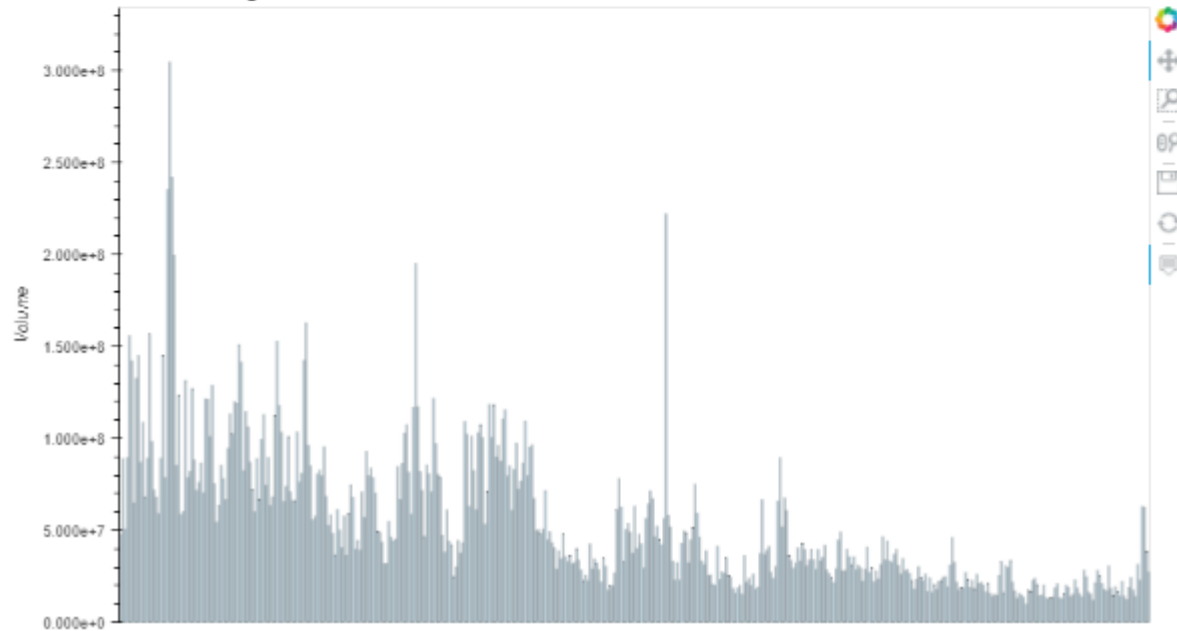
This dashboard presents a visual analysis of Tesla stock (ticker: "TSLA") from January 2020 to October 2021. You can navigate through the tabs below and interact with the plots to explore more details about the Tesla stock.

Historical Data   Candlestick Plot   Closing Price and Volume   Daily Returns

TSLA Closing Price & Simple Moving Average



TSLA Trading Volume

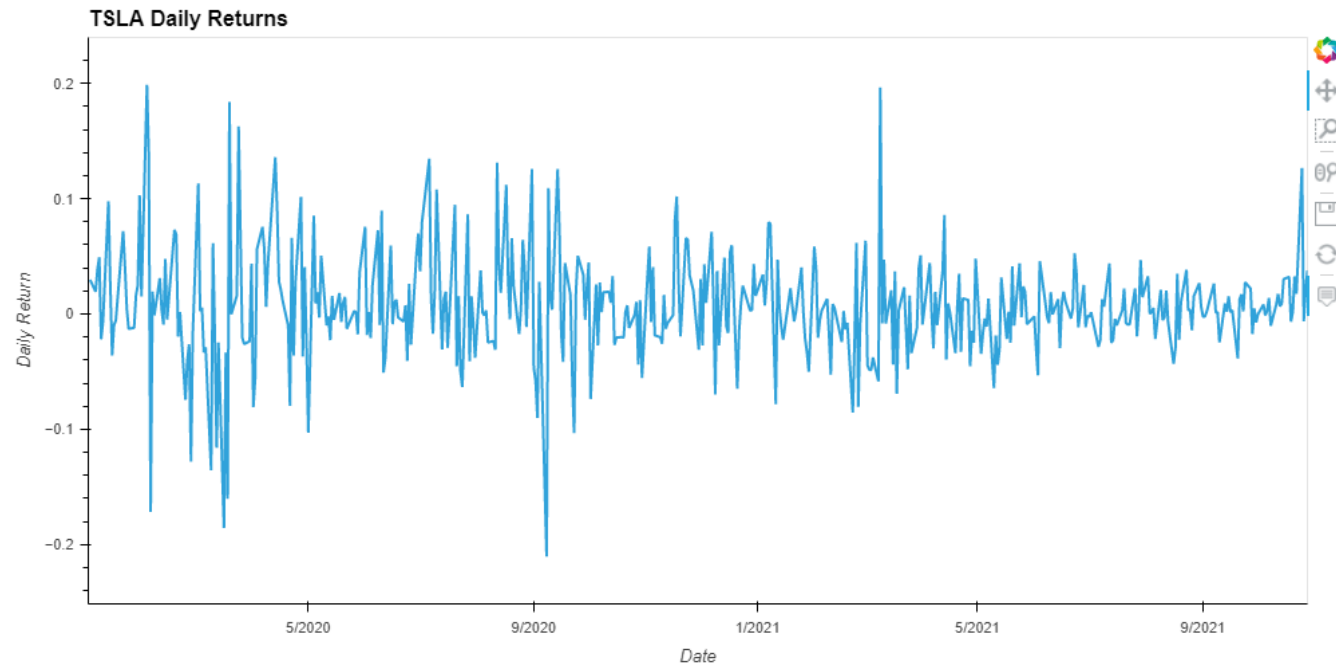


# Daily Returns

## Tesla Stock Analysis Dashboard

This dashboard presents a visual analysis of Tesla stock (ticker: "TSLA") from January 2020 to October 2021. You can navigate through the tabs below and interact with the plots to explore more details about the Tesla stock.

Historical Data   Candlestick Plot   Closing Price and Volume   **Daily Returns**



## Concluding Thoughts

In this article, we have gone over an example of how to use Python for building interactive dashboards. While there are other programs and software available for data visualization purposes (e.g. Power BI, Tableau),

Python's data visualization libraries represent a great resource for people interested in leveraging Python's functionalities.

Regardless of the data visualization program or software of your preference, always be sure to build dashboards easy to understand, that tell a story, provide insights and help business leaders make informed decisions.

. . .

*If you found this article useful, feel welcome to download my personal codes on [GitHub](#). You can also email me directly at [rsalaza4@binghamton.edu](mailto:rsalaza4@binghamton.edu) and find me on [LinkedIn](#). Interested in learning more about data analytics, data science and machine learning applications in the engineering field? Explore my previous articles by visiting my [Medium profile](#). Thanks for reading.*

- Robert

---

**Get an email whenever Roberto Salazar publishes a new article.**

If you like my content, feel free to subscribe and you will get an email every time I publish a new article. Feel free to contact me at [rsalaza4@binghamton.edu](mailto:rsalaza4@binghamton.edu) Thanks for

reading!



Subscribe

Emails will be sent to [mjaworsk@stevens.edu](mailto:mjaworsk@stevens.edu).  
[Not you?](#)

Data Analysis

Data Science

Python

Programming

Software Development

## Learn more.

Medium is an open platform where 170 million readers come to find insightful and dynamic thinking. Here, expert and undiscovered voices alike dive into the heart of any topic and bring new ideas to the surface. [Learn more](#)

## Make Medium yours.

Follow the writers, publications, and topics that matter to you, and you'll see them on your homepage and in your inbox. [Explore](#)

## Write a story on Medium.

If you have a story to tell, knowledge to share, or a perspective to offer — welcome home. It's easy and free to post your thinking on any topic. [Start a blog](#)



[About](#) [Write](#) [Help](#) [Legal](#)