```
import pandas as pd
import cufflinks as cf
cf.go_offline() # Configure Plotly to work in offline mode
import numpy as np
import yfinance as yf
from datetime import date
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
import mercury as mr
from dateutil.relativedelta import relativedelta

app = mr.App(title="📈 Stock Price Dashboard", description="Dashboard with financial data",
show_code=False)
```

from mercury import Select

```
#ticker: This widget presents a selection of available stock tickers

ticker = Select(label="Please select ticker", value='DIS',
                choices=['NVDA', 'DIS', 'AMD', 'TSM','MU', 'INTC'])
```

Error displaying widget: model not found

[11]:

```
mr.Md(f"# Selected ticker: {ticker.value}")
```

Selected ticker: DIS

[26]:

```
#time period for data analysis

period = mr.Numeric(label="Past Month(s)", value=3, min=1, max=12)
```

Error displaying widget: model not found

[13]:

```
# Download Walt Disney Co (DIA) stock data for analysis
```

```
stock_data = yf.download(ticker.value, start=date.today() - relativedelta(months=+period.value),
end=date.today())
stock_data
```

```
[**********100%%**********]  1 of 1 completed
```

[13]:

Date	Open	High	Low	Close	Adj Close	Volume
2023-08-09	89.199997	89.559998	87.040001	87.489998	87.489998	32517900
2023-08-10	89.970001	92.529999	87.750000	91.760002	91.760002	56716800
2023-08-11	91.320000	91.349998	88.860001	89.019997	89.019997	21925700
2023-08-14	88.989998	89.300003	87.989998	88.809998	88.809998	13676400
2023-08-15	87.989998	88.230003	86.889999	87.059998	87.059998	13677900
...	...	...	...	...	...	...
2023-11-02	81.849998	83.300003	81.820000	83.290001	83.290001	11518500
2023-11-03	84.169998	85.930000	84.160004	85.070000	85.070000	11322700
2023-11-06	85.080002	85.800003	83.589996	84.019997	84.019997	14160900
2023-11-07	84.160004	84.680000	83.949997	84.589996	84.589996	10540200
2023-11-08	84.139999	84.919998	83.949997	84.500000	84.500000	21284449

65 rows × 6 columns

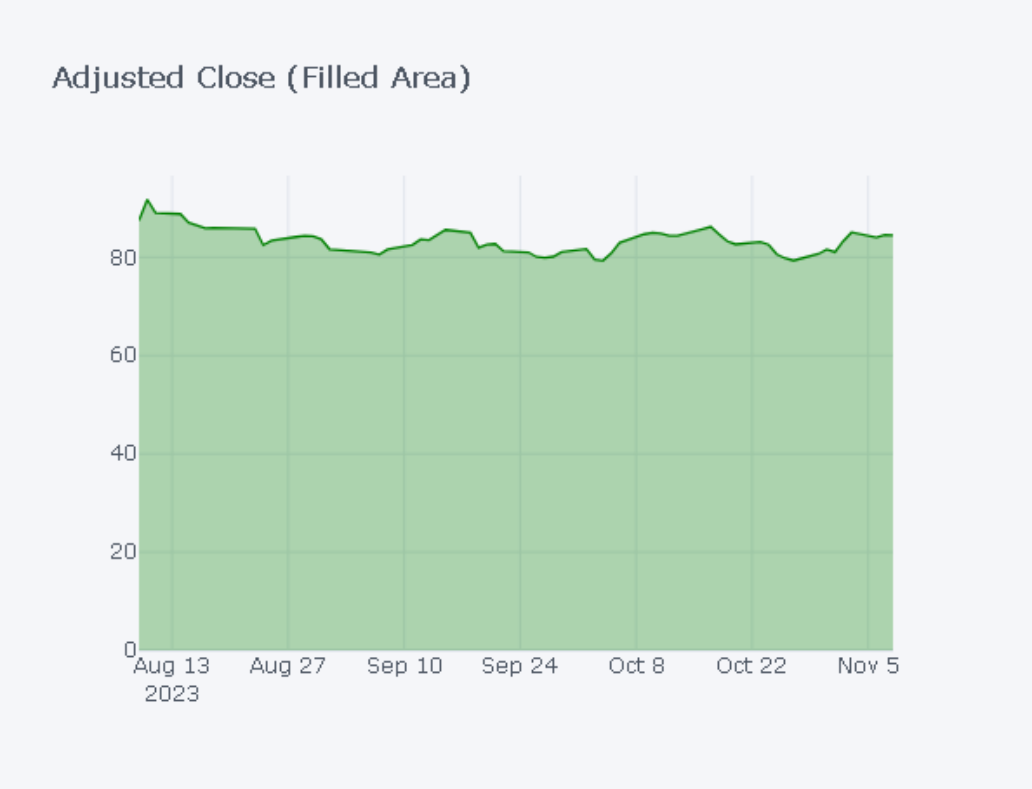
[25]:

```
# Create and display an interactive line plot of DIA's Adjusted Close Prices

stock_data['Adj Close'].iplot(title='Adjusted Close', colors=['green'])
```

[15]:

 # Create and display an interactive filled area plot of DIA's Adjusted Close Prices

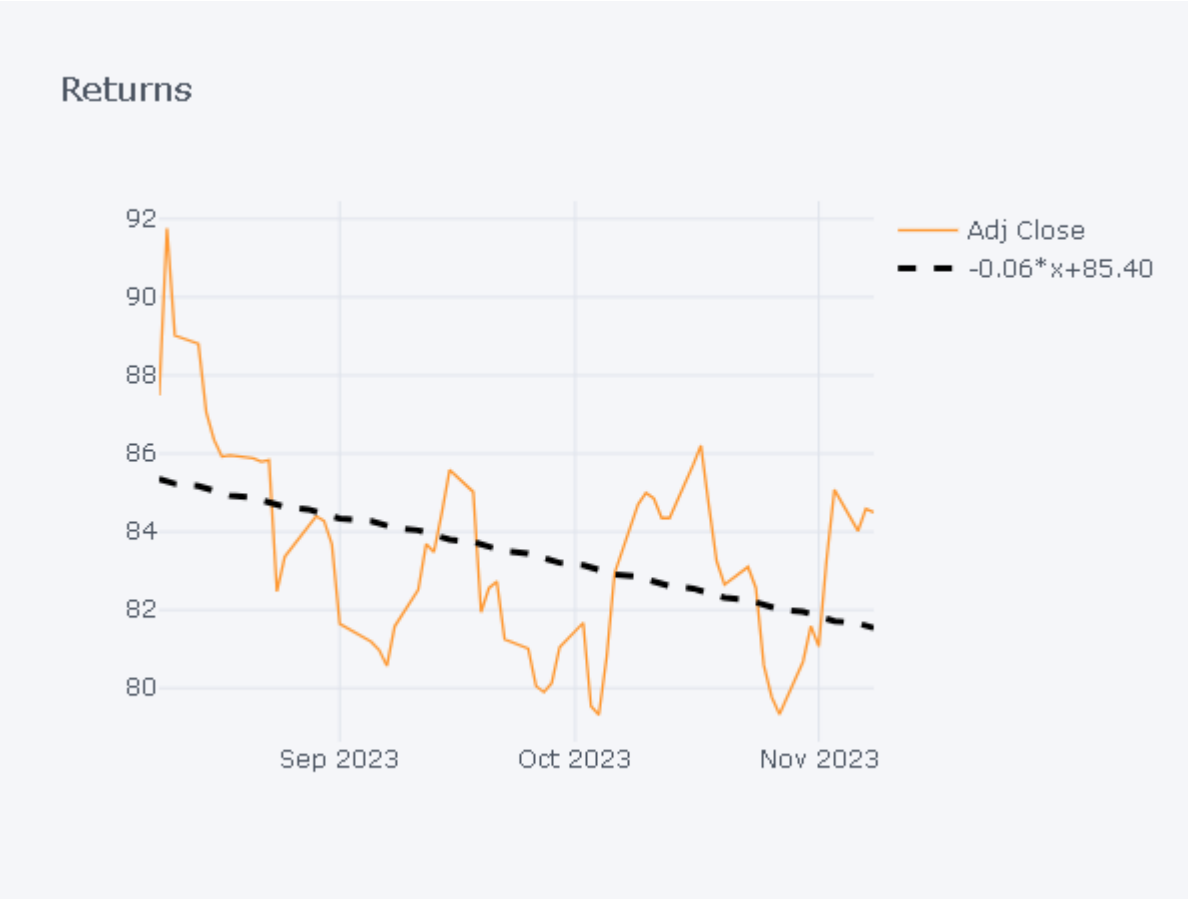
stock_data['Adj Close'].iplot(title='Adjusted Close (Filled Area)', fill=**True**, colors=['green'])



[16]:

 # Create and display an interactive line plot of DIA's Returns with a best-fit line

```
stock_data['Adj Close'].iplot(title='Returns', bestfit=True, bestfit_colors=['black'])
```



[24]:

 # Create a Quantitative Figure (QuantFig) for the DIS stock data

```
qf = cf.QuantFig(stock_data, title='Quantitative Figure', legend='top', name=ticker.value)
qf.add_sma([10, 20], width=2, color=['green', 'lightgreen'], legendgroup=True)
qf.add_bollinger_bands()
qf.add_volume()
qf.iplot()
```

[30]:

```
# Download stock data for semiconductor companies

semiconductor_tickers = ['DIS', 'NVDA', 'INTC', 'AMD', 'TSM','MU']
semiconductor_data = yf.download(semiconductor_tickers, start=date.today() -
relativedelta(months=+period.value), end=date.today())
semiconductor_data
[*********************100%%**********************]  6 of 6 completed
```

[30]:

	Adj Close						Close				...	Open				Volume					
	AMD	DIS	INTC	MU	NVDA	TSM	AMD	DIS	INTC	MU	...	INTC	MU	NVDA	TSM	AMD	DIS	INTC	MU	NVDA	TSM
Date																					
202	110.	87.	34.	66.	425.	93.	110.	87.	34.	66.	...	35.	67.	442.	95.	562	325	287	100	586	642

	Adj Close						Close				Open				Volume					
Date	AMD	DIS	INTC	MU	NVDA	TSM	AMD	DIS	INTC	MU	...	INTC	MU	NVDA	TSM	AMD	DIS	INTC	MU	NVDA	TSM
3-08-09	47.000001	48.999998	16.764648	58.891343	50.494474	49.475959	47.000001	48.999998	27.999999	69.999997		0.000000	77.999999	73.999990	83.000002	57.100000	17.900000	20.200000	43.200000	44.900000	69.000000
2023-08-10	110.230000003	91.760000038	34.563381	65.331017	423.845093	94.310524	110.230000003	91.760000038	34.680000000	65.440002		34.700001	67.059998	421.600006	95.449997	572.978000	56.716800	30.957300	127.753000	492.705000	74.596000
2023-08-11	107.570000000	89.019997	34.775650	64.262794	408.516327	91.515022	107.570000000	89.019997	34.889999	64.370003		34.459999	64.900002	417.510001	93.500000	556.187000	21.925700	28.752000	118.208000	532.009000	106.531000
2023-	111.98	88.80	35.56	68.16	437.49	92.58	111.98	88.80	35.68	68.27		34.66	64.37	404.85	91.69	574.211	13.676	36.896	21.954	69.028	74.897

	Adj Close						Close				...	Open				Volume					
	AMD	DIS	INTC	MU	NVDA	TSM	AMD	DIS	INTC	MU	...	INTC	MU	NVDA	TSM	AMD	DIS	INTC	MU	NVDA	TSM
Date																					
08-14	0003	9998	30061	62883	39588	94477	0003	9998	0000	9999		9998	0003	9985	0002	800	400	100	0000	600	00
2023-08-15	111.34999998	87.059998	34.656044	65.221199	439.363800	91.206627	111.349998	87.059998	34.770000	65.330002	...	35.410000	67.589996	445.600006	92.519997	53816000	13677900	27420300	13808500	67651200	60695000
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2023-11-02	107.830000002	83.290000001	37.576443	70.440002	435.059998	90.480003	107.830000002	83.290000001	37.700000	70.440002	...	37.580002	70.320000	433.279999	90.400002	71377700000	11518500000	35440000000	11926000000	40917200000	11032600000
2022	112.	85.	38.	72.	450.	91.	112.	85.	38.	72.	...	38.	71.	440.	91.	655	113	304	144	423	922

	Adj Close						Close				...	Open				Volume			
	AMD	DIS	INTC	MU	NVDA	TSM	AMD	DIS	INTC	MU	...	INTC	MU	NVDA	TSM	AMD	DIS	INTC	MU	NVDA	TSM
Date																					
3-11-03	25.000000	70.000000	14.999999	58.000002	49.988801	79.000001	25.000000	70.000000	139.999999	58.000002		0.000000	30.000003	200.000112	65.000002	7660000	22700000	9340000	5540000	8550000	38000
2023-11-06	111.750000000	84.019999700	37.950000001	72.919999800	457.510001000	92.599999800	111.750000000	84.019999700	37.950000001	72.919999800	...	38.020000000	72.580000002	452.850000006	92.000000000	49270400	14160900	23877200	10073900	40073300	8402700
2023-11-07	113.449999700	84.589999600	38.770000000	72.800000003	459.549988800	92.410000400	113.449999700	84.589999600	38.770000000	72.800000003	...	38.240000002	72.480000003	457.190000002	92.250000000	50550200	10540200	32052000	8248400	34316500	6241300
2023-	113.580000	84.500000	37.910000	72.270000	465.730000	92.000000	113.580000	84.500000	37.910000	72.270000	...	38.840000	72.620000	461.000000	93.330000	41842000	26552000	33346000	9916200	34585000	6537000

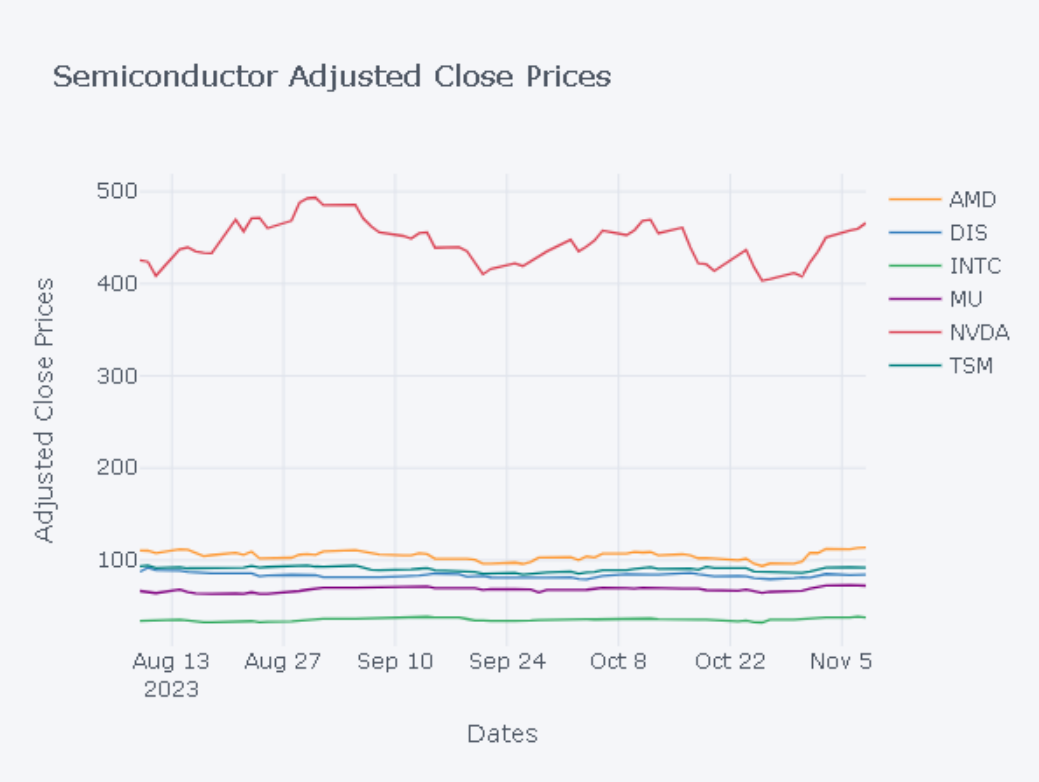
	Adj Close						Close				...	Open					Volume			
Date	AMD	DIS	INTC	MU	NVDA	TSM	AMD	DIS	INTC	MU	...	INTC	MU	NVDA	TSM	AMD	DIS	INTC	MU	NVDA	TSM
11-08	99.96	000.00	99.98	99.99	990.00	000.00	99.96	000.00	99.98	99.99	...	99.98	000.03	000.00	000.02	400	500	700	00	400	00

65 rows × 36 columns

[19]:

 # Create and display an interactive line plot of Adjusted Close Prices for semiconductor companies

semiconductor_data['Adj Close'].iplot(title='Semiconductor Adjusted Close Prices', xTitle='Dates', yTitle='Adjusted Close Prices')

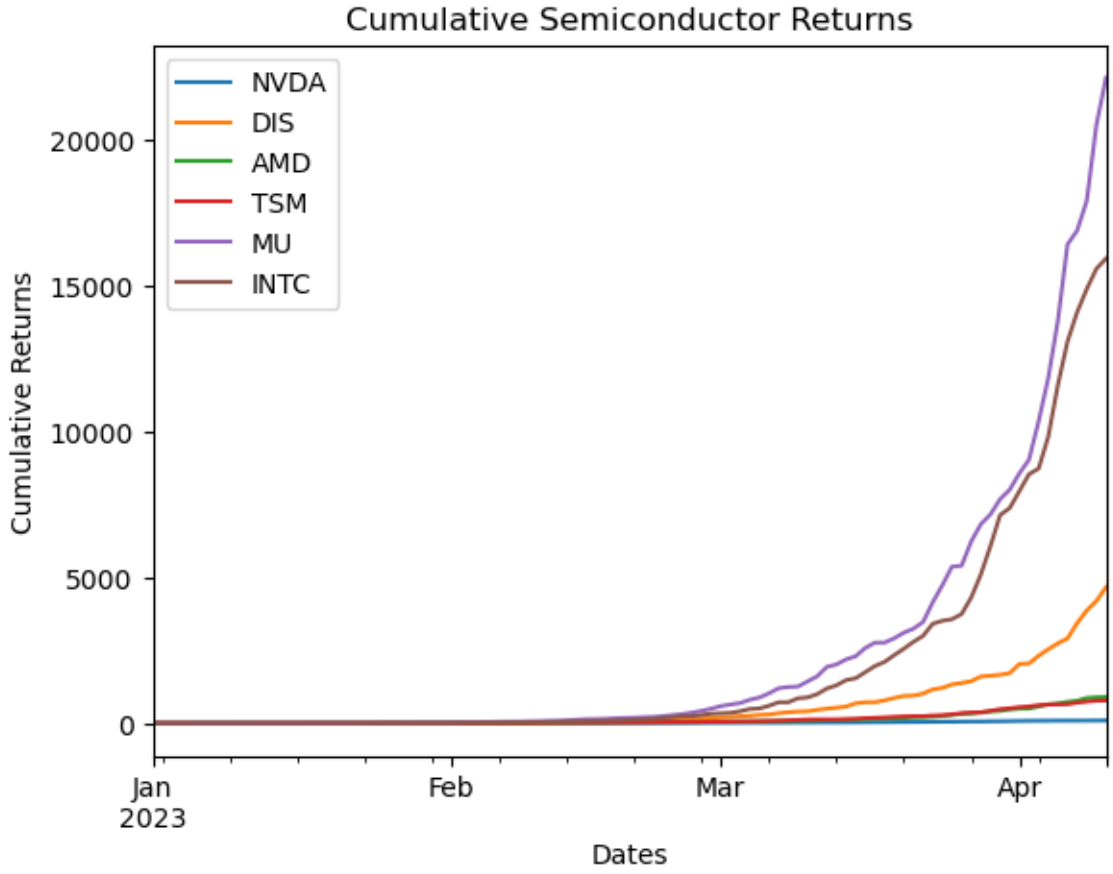


[34]:

```
import numpy as np
import pandas as pd

# Assuming you have 6 companies in the semiconductor_returns DataFrame
num_companies = 6

# Generate example data for demonstration
dates = pd.date_range(start='2023-01-01', periods=100)
semiconductor_returns = pd.DataFrame(np.random.rand(100, num_companies), index=dates,
columns=["NVDA", "DIS", "AMD", "TSM", "MU", "INTC"])
semiconductor_returns.iloc[0, :] = 0  # Set the initial value of returns to 0 (first day)

# Define the weights for each company in the portfolio
weights = np.array([0.1, 0.2, 0.15, 0.15, 0.2, 0.2])

# Calculate the weighted returns for each company
weighted_returns = semiconductor_returns.mul(weights)

# Calculate cumulative returns for the portfolio
cumulative_returns = (weighted_returns + 1).cumprod()

# Plot the cumulative returns
cumulative_returns.plot(title='Cumulative Semiconductor Returns', xlabel='Dates', ylabel='Cumulative
Returns')
```

[34]:

```
<AxesSubplot:title={'center':'Cumulative Semiconductor Returns'}, xlabel='Dates', ylabel='Cumulative Ret
urns'>
```



[ ]: