

CH08_SP500_Python_Supplement

September 3, 2026

1 Chapter 8 — S&P 500 (Python supplement)

Condensed notebook for the **Python** portion of *Chapter 8: S&P 500 Data*.

Target graphics:

- Indexed S&P 500 performance by month since inauguration (**Figure 8.14**; book target **Figure 8.1**)

Data: S&P 500 - Historical Data.csv in Data For Condensed Notebooks.

Dependencies:

- pandas
- numpy
- plotly

1.1 Imports and display options

```
[1]: # Import packages.  
import pandas as pd  
import numpy as np  
import plotly.express as px
```

```
[2]: # Avoid SettingWithCopy warnings.  
pd.options.mode.copy_on_write = True
```

```
/var/folders/_x/nbw2t83x75d0lr3jn4pg_cx00000gn/T/ipykernel_26650/3183164771.py:2  
: Pandas4Warning: The 'mode.copy_on_write' option is deprecated. Copy-on-Write  
can no longer be disabled (it is always enabled with pandas >= 3.0), and setting  
the option has no impact. This option will be removed in pandas 4.0.  
pd.options.mode.copy_on_write = True
```

```
[3]: # Set the maximum number of DataFrame rows to display.  
pd.options.display.max_rows = 8
```

```
[4]: # Set output options.  
import plotly.io as pio  
pio.renderers.default = "pdf+jupyterlab+notebook"
```

1.2 Loading and cleaning the data

```
[5]: df = pd.read_csv('../Data For Condensed Notebooks/S&P 500 - Historical Data.  
      ↪csv')  
df
```

```
[5]:
```

	Date	Close/Last	Open	High	Low	President
0	2/7/25	6025.99	6083.13	6101.28	6019.96	NaN
1	2/6/25	6083.57	6072.22	6084.03	6046.83	NaN
2	2/5/25	6061.48	6020.45	6062.86	6007.06	NaN
3	2/4/25	6037.88	5998.14	6042.48	5990.87	NaN
...	...	...	...	...	...	...
2519	2/12/15	2088.48	2069.98	2088.53	2069.98	NaN
2520	2/11/15	2068.53	2068.55	2073.48	2057.99	NaN
2521	2/10/15	2068.59	2049.38	2070.86	2048.62	NaN
2522	2/9/15	2046.74	2053.47	2056.16	2041.88	NaN

[2523 rows x 6 columns]

```
[6]: df.dtypes
```

```
[6]: Date          str  
Close/Last    float64  
Open          float64  
High          float64  
Low           float64  
President      str  
dtype: object
```

Parse dates; keep only Biden and Trump rows (query).

```
[7]: df['Date'] = pd.to_datetime(df['Date'])  
df = df.query('President == "Biden" | President == "Trump"')  
df
```

```
/var/folders/_x/nbw2t83x75d0lr3jn4pg_cx00000gn/T/ipykernel_26650/3575661020.py:1  
: UserWarning: Could not infer format, so each element will be parsed  
individually, falling back to `dateutil`. To ensure parsing is consistent and  
as-expected, please specify a format.
```

```
df['Date'] = pd.to_datetime(df['Date'])
```

```
[7]:
```

	Date	Close/Last	Open	High	Low	President
14	2025-01-17	5996.66	5995.40	6014.96	5978.44	Biden
15	2025-01-16	5937.34	5963.61	5964.69	5930.72	Biden
16	2025-01-15	5949.91	5905.21	5960.61	5905.21	Biden
17	2025-01-14	5842.91	5859.27	5871.92	5805.42	Biden
...	...	...	...	...	...	...
2020	2017-01-26	2296.68	2298.63	2300.99	2294.08	Trump
2021	2017-01-25	2298.37	2288.88	2299.55	2288.88	Trump

2022	2017-01-24	2280.07	2267.88	2284.63	2266.68	Trump
2023	2017-01-23	2265.20	2267.78	2271.78	2257.02	Trump

[2010 rows x 6 columns]

1.3 Calculations

Hard-code term start dates, compute months since start, average `Open` by president and month, then index to 100 at month 0.

```
[8]: biden_start_date = pd.to_datetime('2021/01/21')
      trump_start_date = pd.to_datetime('2017/01/23')
```

```
[9]: df.loc[:, 'Start Date'] = np.where(df['President'] == 'Biden',
                                         biden_start_date, trump_start_date)
```

```
[10]: month_offset = df['Date'].dt.to_period('M') - df['Start Date'].dt.to_period('M')
      df.loc[:, 'Months Since Start'] = month_offset.apply(lambda offset: offset.n)
      df.head(50)
```

```
[10]:
```

	Date	Close/Last	Open	High	Low	President	Start Date \
14	2025-01-17	5996.66	5995.40	6014.96	5978.44	Biden	2021-01-21
15	2025-01-16	5937.34	5963.61	5964.69	5930.72	Biden	2021-01-21
16	2025-01-15	5949.91	5905.21	5960.61	5905.21	Biden	2021-01-21
17	2025-01-14	5842.91	5859.27	5871.92	5805.42	Biden	2021-01-21
..	...	...	...	...	...	...	...
60	2024-11-08	5995.54	5976.76	6012.45	5976.76	Biden	2021-01-21
61	2024-11-07	5973.10	5947.21	5983.84	5947.21	Biden	2021-01-21
62	2024-11-06	5929.04	5864.89	5936.14	5864.89	Biden	2021-01-21
63	2024-11-05	5782.76	5722.43	5783.44	5722.10	Biden	2021-01-21

	Months Since Start
14	48
15	48
16	48
17	48
..	...
60	46
61	46
62	46
63	46

[50 rows x 8 columns]

Group by president and month; take mean `Open`; merge starting values for indexing.

```
[11]: df_by_month = df.groupby(['President', 'Months Since Start']).agg({'Open': lambda x: x.agg('mean')})
```

```
df_by_month
```

```
[11]:
```

President	Months Since Start	Open
Biden	0	3826.710000
	1	3878.793158
	2	3906.684783
	3	4134.019048
...	...	...
Trump	45	3423.191364
	46	3543.430000
	47	3692.005455
	48	3780.282500

```
[98 rows x 1 columns]
```

```
[12]: biden_start_open = df_by_month.loc['Biden', 0]['Open']
trump_start_open = df_by_month.loc['Trump', 0]['Open']
start_open_df = pd.DataFrame({'President': ['Biden', 'Trump'],
                              'Start Open': [biden_start_open,
                                             ↪trump_start_open]})
df_by_month.reset_index(inplace=True)
df_by_month = pd.merge(df_by_month, start_open_df, on=['President'])
df_by_month['Indexed Open'] = (df_by_month['Open'] / df_by_month['Start Open'])_
↪* 100
df_by_month
```

```
[12]:
```

	President	Months Since Start	Open	Start Open	Indexed Open
0	Biden	0	3826.710000	3826.710000	100.000000
1	Biden	1	3878.793158	3826.710000	101.361043
2	Biden	2	3906.684783	3826.710000	102.089910
3	Biden	3	4134.019048	3826.710000	108.030633
..	...	...	...	...	...
94	Trump	45	3423.191364	2283.174286	149.931233
95	Trump	46	3543.430000	2283.174286	155.197526
96	Trump	47	3692.005455	2283.174286	161.704933
97	Trump	48	3780.282500	2283.174286	165.571351

```
[98 rows x 5 columns]
```

Figure 8.14 — Indexed performance line chart (solid Biden, dashed Trump).

```
[13]: px.line(df_by_month,
              x = 'Months Since Start',
              y = 'Indexed Open',
              color = 'President',
              line_dash= 'President',
              line_dash_map={"Biden": "solid", "Trump": "dash"},
```

```
title = 'S&P 500 Performance - Biden vs Trump (1st Term)')
```

S&P 500 Performance - Biden vs Trump (1st Term)

