Basic timeseries analysis in Pandas

Stelios Christodoulou

PyData London

June 2025

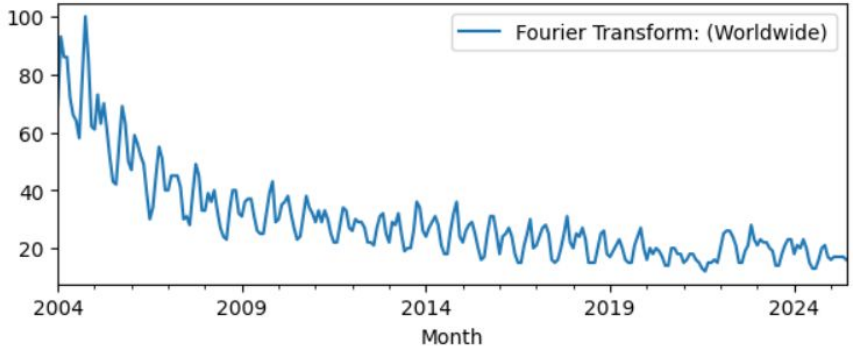
Basic timeseries analysis in Pandas

autocorrelation_plot()	lag_plot()	.diff()	.autocorr()
Test for randomness and periodicity	Focus on a particular lag identified from the autocorrelation plot	Difference function: highlighting discontinuities, removing “dc components”	Calculate the numerical value for autocorrelation at the particular lag

Autocorrelation Plot

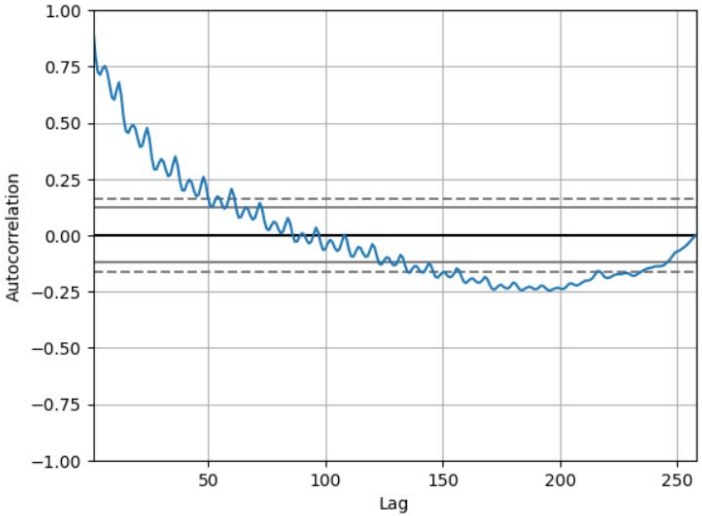
```
[26]: trends_df.plot(figsize=[7,2.5])
```

```
[26]: <Axes: xlabel='Month'>
```




```
[28]: pd.plotting.autocorrelation_plot(trends_df)
```

```
[28]: <Axes: xlabel='Lag', ylabel='Autocorrelation'>
```

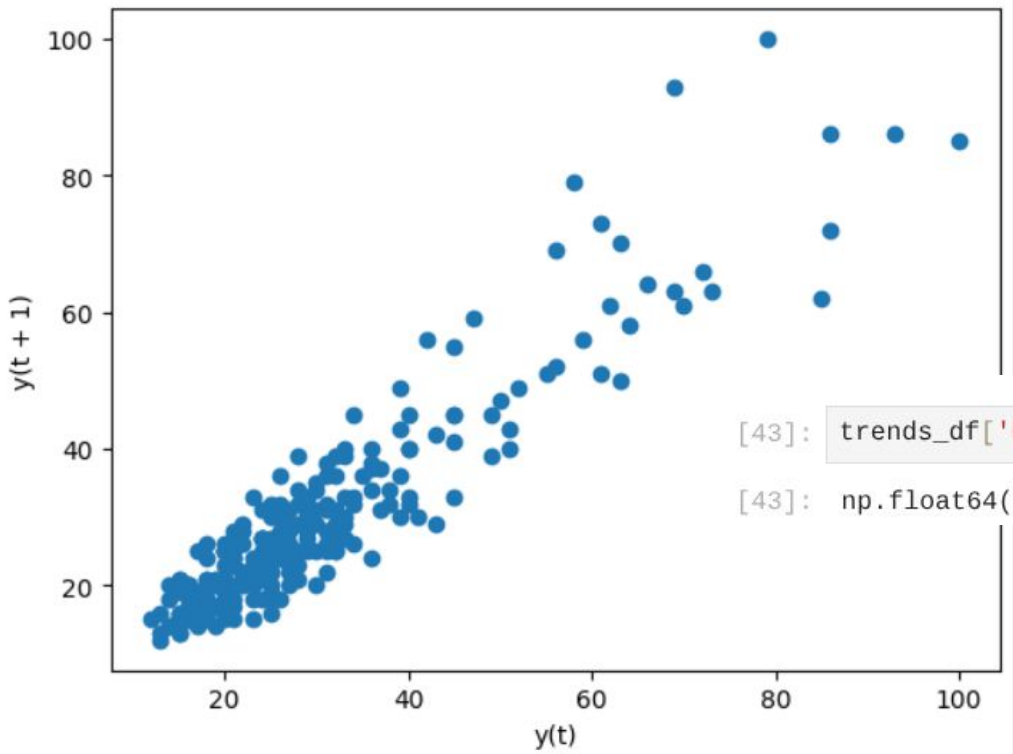


- Not random
- Trend dominates over periodicity

Lag 1 and autocorr

```
pd.plotting.lag_plot(trends_df,lag=1)
```

```
<Axes: xlabel='y(t)', ylabel='y(t + 1)'>
```




```
[43]: trends_df['Fourier Transform: (Worldwide)'].autocorr(lag=1)
```

```
[43]: np.float64(0.9293125693134024)
```

- Not random confirmed

For the computationally minded

```
[31]: (trends_df
       .assign(Fourier_Transform_shifted_1m=trends_df
               .shift(periods=1)
              )
       .corr()
       .loc['Fourier_Transform_shifted_1m','Fourier Transform: (Worldwide)']
      )

[31]: np.float64(0.9293125693134031)

[43]: trends_df['Fourier Transform: (Worldwide)'].autocorr(lag=1)

[43]: np.float64(0.9293125693134024)
```

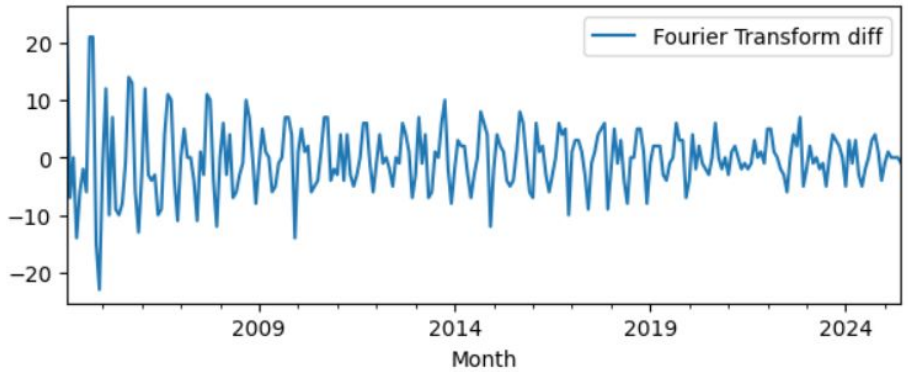
- Why are the numbers identical?
- Which one is correct?

Diff and Lag

```
def non_partial_diff(df):
    return (df
            .diff()
            .rename(columns={"Fourier Transform: (Worldwide)":"Fourier Transform diff"})
            .dropna()
           )
```

```
[33]: trends_df.pipe(non_partial_diff).plot(figsize=[7,2.5])
```

```
[33]: <Axes: xlabel='Month'>
```

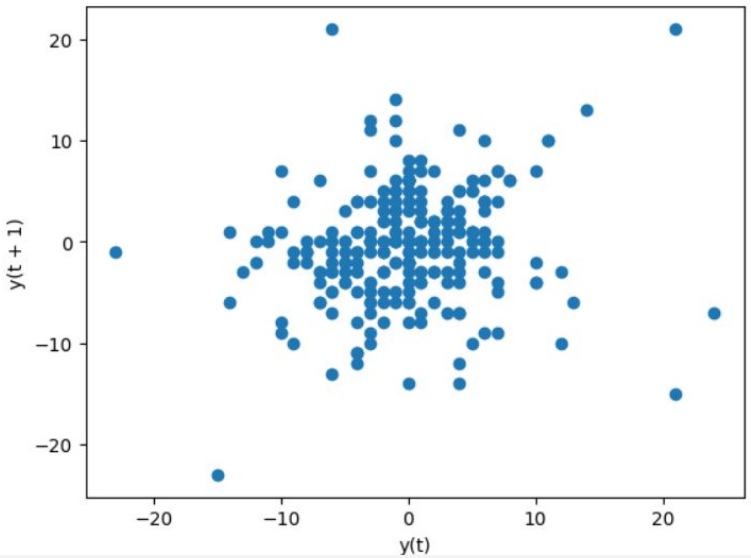



```
trends_df.pipe(non_partial_diff)['Fourier Transform diff'].autocorr(lag=1)
```

```
np.float64(0.16995405644243017)
```

```
pd.plotting.lag_plot(trends_df.pipe(non_partial_diff),lag=1)
```

```
<Axes: xlabel='y(t)', ylabel='y(t + 1)'>
```

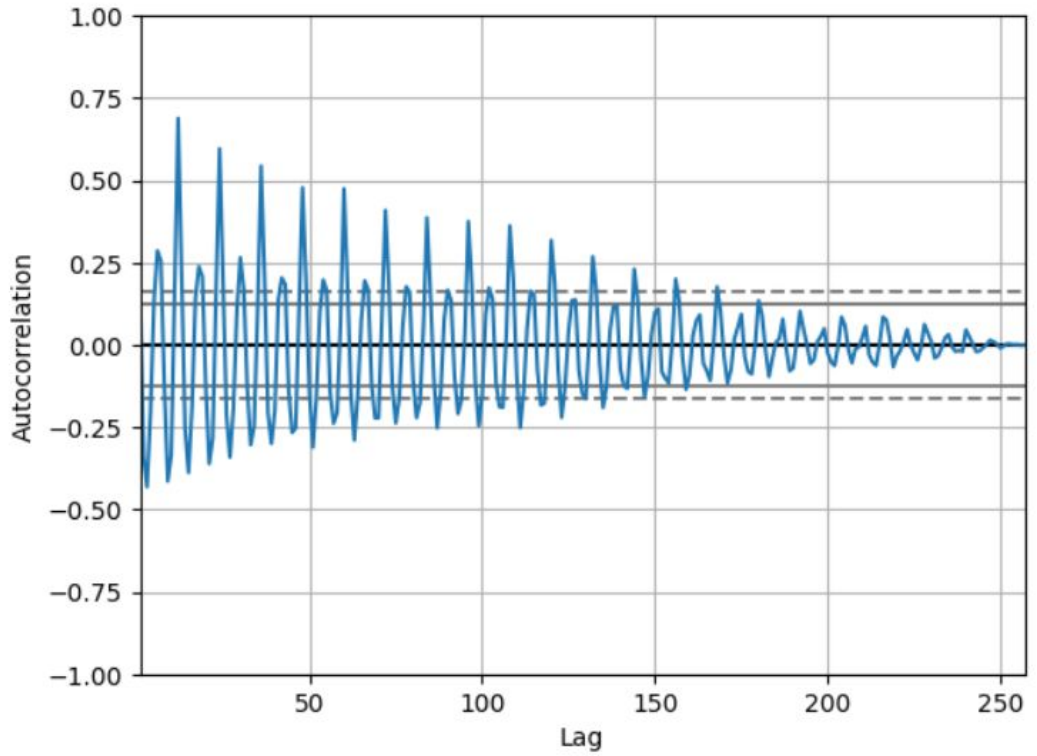


- Diff makes lag 1 random

Diff and Autocorrelation

```
pd.plotting.autocorrelation_plot(trends_df.pipe(non_partial_diff))
```

```
<Axes: xlabel='Lag', ylabel='Autocorrelation'>
```

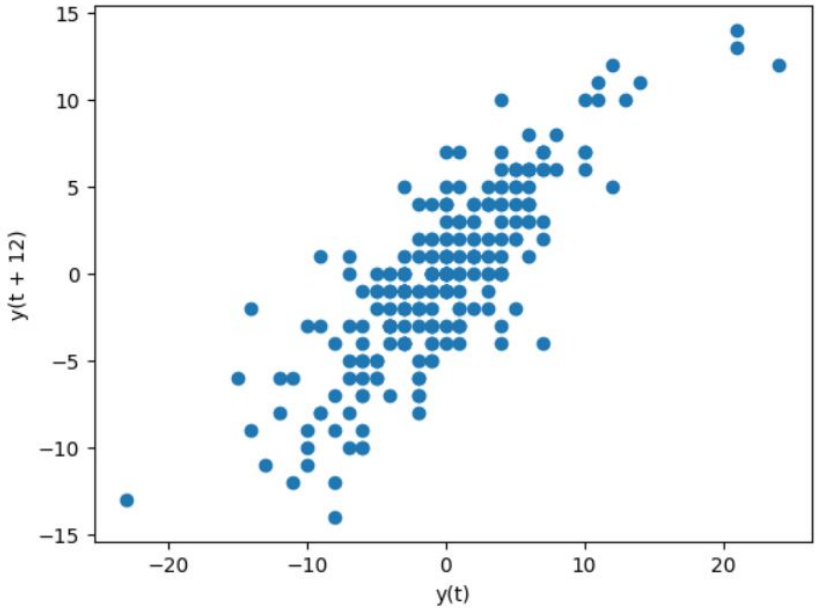


- Periodicity comes to the fore

Lag for periodicity

```
pd.plotting.lag_plot(trends_df.pipe(non_partial_diff),lag=12)
```

```
<Axes: xlabel='y(t)', ylabel='y(t + 12)'>
```



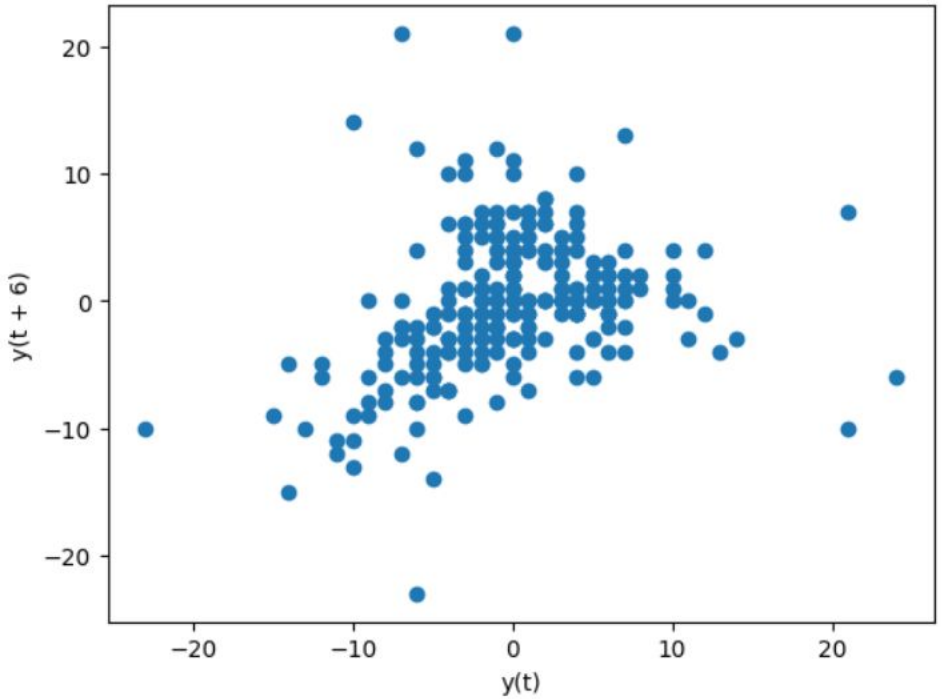

```
trends_df.pipe(non_partial_diff)['Fourier Transform diff'].autocorr(lag=12)
```

```
np.float64(0.8195626269703368)
```

- Diff and lag 12 quite correlated

```
pd.plotting.lag_plot(trends_df.pipe(non_partial_diff),lag=6)
```

```
<Axes: xlabel='y(t)', ylabel='y(t + 6)'>
```




```
trends_df.pipe(non_partial_diff)['Fourier Transform diff'].autocorr(lag=6)
```

```
np.float64(0.30112376919812994)
```

- Diff and lag 6 weakly correlated

Thank you

Bibliography
- Pandas documentation
- NIST/SEMATECH e-Handbook of Statistical Methods
- Random Processes chapter in Communication Systems (Haykin)

https://github.com/stelios-c/sig_proc/

Inspired from a Linked In post by Kunpeng (KP) Liao.